AN ADAPTIVE FILTERING APPROACH TOWARD SPEECH ENHANCEMENT

Ronald Howell Frazier

AN ADAPTIVE FILTERING APPROACH TOWARD SPEECH ENHANCEMENT

by

Ronald Howell Frazier

B.S., U.S. Coast Guard Academy

(1971)

SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREES OF

ELECTRICAL ENGINEER

AND

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1975

Signature of Author ....
Department of Electrical Engineering, May 9, 1975

Certified by ....
Thesis Supervisor

Accepted by ....
Chairman, Departmental Committee on Graduate Students

AN ADAPTIVE FILTERING APPROACH TOWARD SPEECH ENHANCEMENT

by

Ronald Howell Frazier

Submitted to the Department of Electrical Engineering on 9 May 1975 in partial fulfillment of the requirements for the Degrees of Electrical Engineer and Master of Science.

ABSTRACT

The use of a digital comb filter for the separation of two speakers was formulated in previous efforts. An adaptive pitch synchronous filter has been developed as an alternative. This development stems from the characteristics and structure of the speech waveform. In the comb filtering development, there were tradeoffs between desired speaker distortion and undesired speaker separation. In the development of the adaptive filtering techniques, the results of these tradeoffs will be examined for possible improvements over the comb filtering techniques.

A series of tests performed on test signal inputs compares the performance of both the comb and adaptive filtering systems. Included in an appendix are several of the main computer programs used in the computer implementation. A series of listening tests will be performed with these systems in future work making the appendix a necessary part of the thesis.

THESIS SUPERVISOR: Siamak Samsam

TITLE: Visiting Scientist, Research Laboratory of Electronics

ACKNOWLEDGEMENTS

I would like to express my gratitude to the Research Laboratory of Electronics for making the necessary computer time available. The financial support provided by the United States Coast Guard deserves a special acknowledgement. The help of Professor Alan V. Oppenheim and other members of the Digital Signal Processing Group at M. I. T. throughout the course of this thesis was also greatly appreciated.

I also wish to sincerely thank Professor Louis Braida for the help and encouragement that he provided especially in the preparation of the written report. Professor Braida made it possible for the PDP-11/45 Computer to be used for the work of this thesis. Other persons whose help during the course of the thesis was greatly appreciated were Nathaniel Durlach, Bruce Hicks, Philip Herman, Michael Portnoff, and William Rabinowicz. I wish to thank Beryl Bergen for her help in the preparation of the thesis.

I also wish to thank Doctor Siamak Samsam for serving as advisor to this thesis and for providing helpful advice on many occasions. I wish to recognize the support and guidance of my parents that ultimately made this work possible. Finally, I especially wish to thank my wife, Karen, whose love, support, and encouragement helped immeasurably throughout the course of my studies at M. I. T.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF FIGURES

LIST OF FIGURES
(Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

CHAPTER I

INTRODUCTION

1.1 Introduction and Goals

The problem of separating a signal waveform from a noise waveform has been the topic of unlimited research for many years. In this discussion the problem of interest is one of separating a speech waveform from a "noise waveform" which may be the speech waveform from a competing speaker. Two methods for speech enhancement are developed in this thesis: comb filtering and adaptive filtering.

The remainder of Chapter I describes the speech waveform and its characteristics in a manner that will be helpful for terminology and modeling. In Chapter II a review of the previous systems dealing with this specific problem will be covered, and the problem formulation introduction for this thesis will be outlined. Chapter III deals with the algorithm that was used in determining the fundamental frequency of a speech waveform including some of the background research that was developed on this method. In Chapter IV the adaptive filtering methods are formulated along with a discussion of the rules used and problems that should be encountered with the system. Chapter V concentrates on the test signal formulation, processing, and results for the different filtering systems implemented. This chapter also deals with the comparison methods that will be used between the outputs of the

different systems. The computer implementation of the systems will be discussed in Chapter VI , and Chapter VII describes the results of the actual speech waveforms that were processed. Included in this chapter are spectrograms of the inputs and outputs for the systems used. The overall results and conclusions are covered in Chapters VIII and IX respectively. The Appendices contain computer programs, flowcharts, and documentation for the purpose of making the continuing work in this problem easier for those persons concerned.

The goals of this thesis can be summarized in the following manner: First, the work that had previously been done was to be duplicated. This involved implementing the computer system described by Vaden Shields.[1] The purpose of this work was to arrive at a starting location before any other work was begun. The system was also implemented for the purpose of comparison with any systems that were generated by future work. Second, a pitch synchronous adaptive filtering scheme was developed and implemented on the computer. Now with these two completely separate systems, their results could be compared to give the third goal. Finally, the ideas of comb filtering and adaptive filtering were examined in order to find a limit to their effectiveness on the speech enhancement problem.

1.2 The Speech Waveform and its Characteristics

Before the speech waveform can be processed by any type of system, the structure, characteristics, and properties of the waveform must be understood. A thorough understanding of these items along with the definition of several terms is necessary before the various processing systems are discussed.

Speech production takes place in an area of the body referred to as the vocal tract. The vocal tract can be best described as an acoustical tube of nonuniform cross-sectional area that originates at the vocal cords and terminates at the lips. The vocal tract of the average male is 17 centimeters long, and is shorter in the average female adult. The nasal tract which lies between the velum and nostrils is another nonuniform cross-sectional area tube that may be acoustically coupled to the vocal tract by the opening at the velum.

Within the vocal and nasal tracts lie the major components for speech production. The articulators; composed of the lips, jaws, tongue, and velum vary with the size and shape of the tract.[2] The velum is also referred to as the soft palate and controls the acoustical coupling between the vocal and nasal tracts. The vocal cords lie at the lower end of the vocal tract. The vocal cords consist of a pair of lips made of ligament and muscle, and the opening between the vocal cords is referred to as the glottis. The words subglottal and supraglottal that are used frequently in the literature refer respectively to the areas immmediately below and above the glottal opening.

In general there are three methods of producing speech waveforms. First and most important, voiced sounds are produced by exciting the vocal tract with quasi-periodic pulses of air that originate in the lungs as a steady flow and are chopped into pulses by the vocal cord vibrations. Second, the vocal tract is constricted in one area, and then by means of forcing the air from the lungs through the constriction, a turbulent air flow is created. Fricative sounds are produced in this manner. Plosive sounds are created by closing the vocal tract momentarily, building up a pressure, and then, releasing it.[3]

Usually sounds that use the vocal cord vibrations for excitation are referred to as voiced, and those that do not use the vocal cords are referred to as unvoiced. The sounds that are produced with the use of the nasal tract are named nasal sounds. For these types of sounds the nasal tract is coupled to the vocal tract at the velum.

The terms pitch, pitch period, and fundamental frequency can all be defined at the same time. These terms have been defined in many ways by different authors. G. Fant defines them in them in the following manner:

> The basic property of a vocal cord sound is its periodicity expressed by the duration T_o of a complete voice period or by the inverse value of the voice fundamental frequency.
>
> $$F_o = 1/T_o$$
>
> Fundamental pitch and fundamental frequency are not synonymous, but these terms can be used interchangeably due to the close one-to-one correspondence. In more strict terminology pitch is a tonal sensation and frequency a property of the sound stimulus. The duration of a pitch cycle always varies somewhat from one period to the next. Such variations are systematic determining the intonation in part, accidental rather than unintentional, but nevertheless of importance for the naturalness of human speech. Only speaking machines are capable of producing a perfectly monotonic pitch.[4]

The fundamental frequency of the vocal cords is directly proportional to their tension, mass, and the subglottal air pressure. Since most adult males have vocal cords that are longer and thicker than adult females the fundamental frequency in the male voice is lower than in the female voice.

For a normally speaking voice the fundamental frequency can range between 60 hertz and 400 hertz. H. L. Shaffer[5], through a series of

experiments, determined that the average fundamental frequency of a male speaker is 125 hertz, while the average for a female is 192 hertz.

Another important term, formant, is defined as the natural frequency of the vocal tract that corresponds to a resonance or peak of energy at various frequencies referred to as resonant frequencies. The resonant or formant frequencies depend on the shape of the vocal tract and the positions of the articulators.

There have been several models proposed for the speech production apparatus. The model proposed by Oppenheim and Schafer[6] gives a good insight into the manner in which the speech waveform is produced.

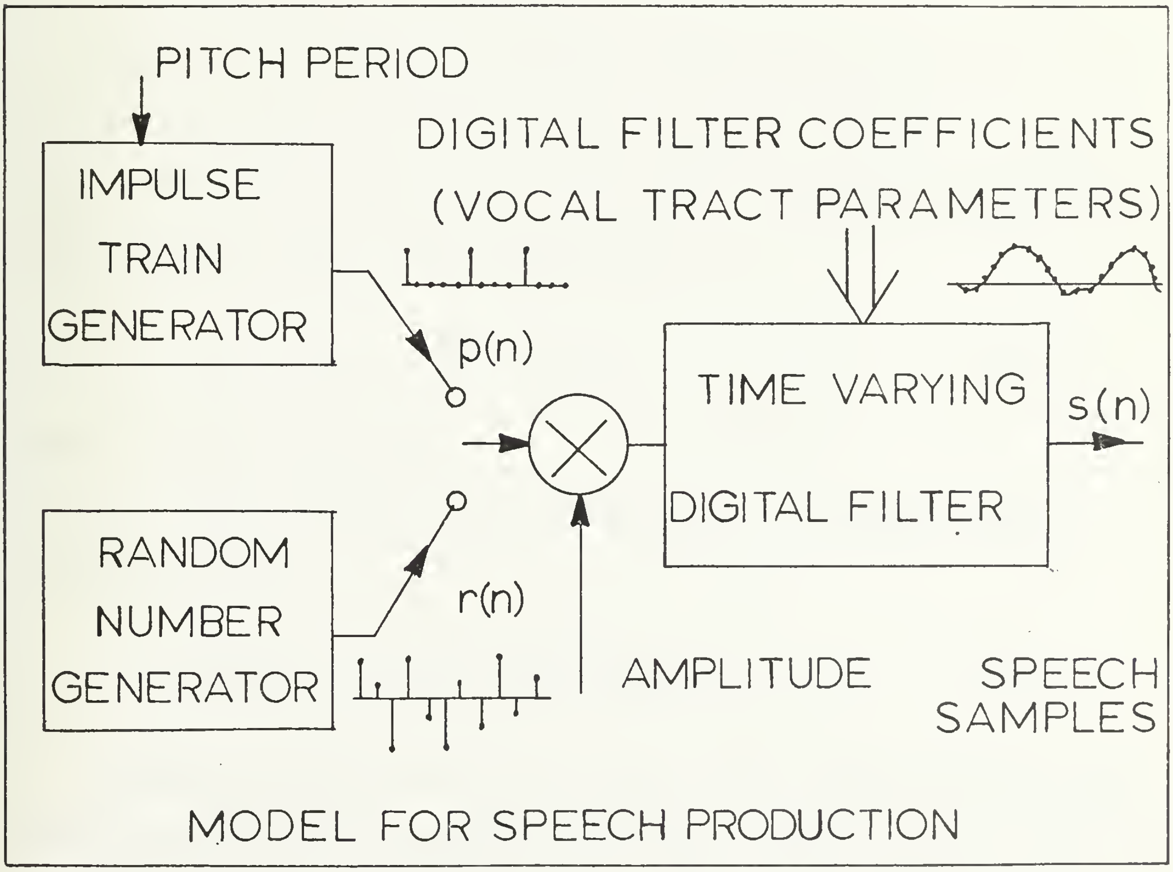

PITCH PERIOD
IMPULSE TRAIN GENERATOR
DIGITAL FILTER COEFFICIENTS (VOCAL TRACT PARAMETERS)
p(n)
TIME VARYING DIGITAL FILTER
s(n)
RANDOM NUMBER GENERATOR
r(n)
AMPLITUDE
SPEECH SAMPLES
MODEL FOR SPEECH PRODUCTION

Figure 1-1[7]

The discrete-time model in figure 1-1 can be examined in several sections. The time varying digital filter corresponds to the vocal tract, and the coefficients of the filter may be changed to correspond to the frequency response of the time varying characteristics of the vocal tract. The vocal tract is usually slowly varying in normal speech and can be considered fixed for short periods of time (on the order of 10 milliseconds). The filter can be excited by a train of impulses, p(n), or by samples from a random number generator, r(n). In voiced speech the impulse train, p(n), is used to excite the filter, and the spacing between the samples corresponds to the pitch period. For unvoiced speech the sequence r(n) is used as a noise-type source for filter excitation.

Oppenheim and Schafer[8] describe a factor g(n) that accounts for the fact that the actual glottal pulses are not impulses. This additional factor can be used with the other two components to describe the speech waveform, $s_v(n)$, for voiced speech.

$$s_V(n) = p(n) * v_V(n) * g(n) \qquad (1.1)$$

where: * denotes convolution

The above expression can also be expressed in its frequency domain representation as:

$$S_V(e^{j\omega}) = P(e^{j\omega})\, V_V(e^{j\omega})\, G(e^{j\omega}) \qquad (1.2)$$

As described above the impulse train, p(n), has a spacing between pulses T_p corresponding to the pitch period. $P(e^{j\omega})$ is also a train of impulses separated by intervals of $fp = 1/Tp$. This shows that the energy of the speech waveform will lie in narrow bands centered about

the fundamental frequency and its harmonics.

Whenever the speech is unvoiced, the speech waveform can be described as:

$$s_u(n) = r(n) * v_u(n) * g(n) \qquad (1.3)$$

Alternatively as before:

$$S_u(e^{j\omega}) = R(e^{j\omega})\ V_u(e^{j\omega})\ G(e^{j\omega}) \qquad (1.4)$$

There is not much that can be said about the properties of $r(n)$ and $R(e^{j\omega})$ other than it has "noise-like" characteristics. Therefore, for unvoiced sounds the spectrum of the signal, $s_u(n)$, lacks any type of harmonic structure.

There are certain other aspects of the human speech system that should be considered. The ear which is considered to be a part of the overall speech system is insensitive to errors in phase of a signal, while on the other hand, the ear is extremely sensitive to errors in the pitch epochs or pitch periods. Many experiments have shown that the pitch of a speech waveform provides its naturalness, and if there are any errors in pitch of a processed signal, it becomes immediately apparent.

This discussion has been directed toward the explanation and definition of certain aspects of the speech waveform that will be used and exploited in the following chapters. It was intended to show how the speech waveforms are produced and how these waveforms can be modeled.

CHAPTER II

REVIEW OF THE LITERATURE

2.1 Shields' System

Since the foundations of this thesis were taken from another thesis, some of the fundamental concepts and ideas used in a thesis by Vaden Shields will be covered briefly. Shields in his thesis, "Separation of Added Speech Signals By Digital Comb Filtering", suggests that the harmonic structure of speech may be the key to the effective removal of additive noise. He attacks the problem in the following manner. First, he looks into the problem of choosing an optimum unit sample response for the filter. Second, he examines the effects of unvoiced speech and filter interaction. Finally, he looks into the tracking and detection of one signal's fundamental frequency from a combined signal. The first two areas will be covered in this summary.

The filtering strategy used is simple. Since voiced speech has its spectral energy concentrated in narrow harmonic bands, another speech signal with the same type of spectrum would have its spectral energy concentrated in narrow harmonic bands that would, for the most part, not overlap those bands of the first signal.[9] This property suggested the use of a comb filter adjusted to allow the first signal to pass and the second to be attenuated.

The comb filter must have several properties when it is used in

this application. First, the filter must be time variant. This property evolves from the fact that the pitch period of the speech varies with time, and the passbands of the filter must be able to move. Second, the filter can only act over a limited portion of the speech waveform at one time. This is due to the fact that the approximation of periodicity of voiced speech only holds for a short time segment. Also, the duration of the impulse response of the filter should be finite and short compared to changes in the speech waveform.[10]

The property above that requires the impulse response of the filter be finite implies that the digital filter be nonrecursive or FIR. There are several ways to design an FIR filter and the method Shields uses is to truncate the infinitely long unit sample response with a finite duration data window. If a filter were desired that would pass only certain frequencies, the "teeth" or passbands would need to be very narrow. One type of filter for this application would be a unit sample train in frequency with the spacing of the samples $f_p = 1/MT$ hertz for $-\infty < f < \infty$ where M is an integer and T is the sampling rate.[11] The unit sample response of this filter is a unit sample train in time with each sample separated by (M - 1) zeros for $-\infty < n < \infty$. This filter would be set to pass only harmonics of f_p.[12]

As mentioned above, the infinitely long unit sample response of the filter must be truncated in order to create the FIR filter. This is done by multiplying the infinitely long unit sample response of the filter by the unit sample response of the data window. The multiplication that the above operation implies in the time domain corresponds to a convolution of the respective transforms in the frequency domain.

The effect of this operation is to create narrow harmonically spaced passbands and this is referred to as a digital comb filter.

The shape of the passbands and the stopbands of the comb filter are a function of the type of window that is used.[13] Shields explored the effects of four common window functions: the rectangular, the Hanning, the Hamming, and the Blackman.

Another way of looking at what is happening in this process is to go back to the step where the filter's infinite duration unit sample response and the data window were being multiplied together. Looking at these results gives the following equations:

Suppose x(n) is a finite duration sequence with a Fourier Transform, $X(e^{j\omega})$, where:

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x(n)\ e^{-j\omega n} \qquad (2.1)$$

Now, suppose that x(n) is multiplied by a function f(n), whose unit sample response is of the following form:

$$f(n) = \frac{1}{m} \sum_{r=0}^{m-1} e^{\frac{j2\pi rn}{m}} \quad ; \text{ for } -\infty < n < \infty \qquad (2.2)$$

f(n) is a train of unit samples that have (m-1) zeros between each unit sample. If z(n) if formed from the product of x(n) and f(n), then, the resulting Fourier Transform of z(n) is calculated as follows:

$$z(n) = x(n)\ f(n) \qquad (2.3)$$

Substituting into equation (2.1)

$$Z(e^{j\omega}) = \sum_{n=0}^{N-1} x(n)\ f(n)\ e^{-j\omega n} \qquad (2.4)$$

Substituting equation (2.2) in for f(n) in (2.4) gives:

$$Z(e^{j\omega}) = \sum_{n=0}^{N-1} x(n) \left[\frac{1}{m} \sum_{r=0}^{m-1} e^{\frac{j2\pi rn}{m}} \right] e^{-j\omega n} \qquad (2.5)$$

Performing an interchange of the summations in (2.5) gives:

$$Z(e^{j\omega}) = \frac{1}{m} \sum_{r=0}^{m-1} \sum_{n=0}^{N-1} x(n)\, e^{-j(\omega - \frac{2\pi r)n}{m}} \qquad (2.6)$$

Equation (2.6) may also be written in the following form:

$$Z(e^{j\omega}) = \frac{1}{m} \sum_{r=0}^{m-1} X\{e^{j(\omega - \frac{2\pi r)}{m}}\} \qquad (2.7)$$

Interpreting (2.7) may be easier if considered in the following form:

$$Z(e^{j\omega}) = \frac{1}{m} \Big\{X(e^{j\omega}) + X(e^{j(\omega-\frac{2\pi)}{m}}) + \ldots$$

$$+ X(e^{j(\omega - \frac{2\pi(m-1))}{m}}) \Big\} \qquad (2.8)$$

From equation (2.8) it can be seen that if $X(e^{j\omega})$ is periodic on the interval $\{0,2\pi\}$, then, $Z(e^{j\omega})$ is composed of a sequence of teeth that are periodic on $\{0,\frac{2\pi}{m}\}$, and of the same form as $X(e^{j\omega})$.

The difference equation that is implemented by this method can be expressed as:

$$y(nT) = \sum_{k=-K}^{K} a_k\, x(nT + kmT) \qquad (2.9)$$

It can be seen that the output is merely a weighted sum of the input values separated by mT seconds. The coefficients, a_k, are determined before the processing starts and are fixed by the input parameter K. This dependency only on the value of K may be shown in the following manner.

Using a Hamming Window with length, N = LM, where L = 2K + 1, and L = the number of coefficients used, the expression for the window is:

$$w(n) = 0.54 - 0.46 \cos (2\pi n/N) \text{ for } 0 \leq n \leq N - 1$$
$$= 0 \qquad \text{elsewhere.} \qquad (2.10)$$

Substituting in for the equivalent expression for N:

$$w(n) = 0.54 - 0.46 \cos (2\pi(iM)/LM) \qquad (2.11)$$

where n = iM are the only points that have non-zero values. Now,

$$w(iM) = 0.54 - 0.46 \cos (2\pi i/L) \qquad (2.12)$$

It can be seen that the coefficients have no dependency on the value of M or fundamental frequency. This shows how the coefficients may be calculated and stored before processing and are always the same after the parameter K has been chosen.

There are several factors or properties of windows that must be considered. First, the longer the unit sample response of the window, the narrower the passbands of the comb filter will be. This occurs when the fundamental frequency decreases. Another consideration is the fact that a limit exists for the duration of the window due to the changing pitch period. For a fixed window length, the choice of window will change the filter characteristics such as; passband width

and stopband attenuation.[14] In his thesis, Vaden Shields studied the choices for type of window and window length.

Both of these studies were experimental. The length of the window was directly proportional to the parameter K and signals were processed with and without an additive noise signal to determine the effects of distortion and separation of the original signal.

The experimental observations agreed with the theoretical results. As the value of K was increased (this corresponds to a longer window length, more coefficients, and narrower passbands), the separation of two speakers improved, but the desired signal became more distorted. Shorter window lengths yielded less separation and less distortion. Through a series of listening experiments the value K = 3 was chosen as a good compromise value from the distortion and separation viewpoints.[15]

The search for the optimum window type started with the immediate elimination of the rectangular window. The rectangular window introduced significantly more distortion than Hamming, Hanning, or Blackman windows. The differences between these three windows were slight, and after listening tests the Blackman window was chosen as the best overall.[16]

The second area of Shields' thesis dealt with the treatment of unvoiced speech. His first approach was to turn off the comb filter upon the detection of a pitch period greater than 20 milliseconds. This approach seemed to work effectively when used on a signal without any noise, but did not effectively separate the desired signal from the noise. During unvoiced sections of speech, the noise signal would "pop out"

of the processed signal. This effect proved to be distracting to listeners and caused an impairment to intelligibility.[17]

Another approach for the treatment of the unvoiced speech segments was used. Whenever an unvoiced segment was detected, the comb filter would continue to use the last valid pitch period. This approach achieved better results for noise suppression, but it tended to distort the desired signal.[18] Shields suggested that other methods or approaches for the treatment of unvoiced sections should be investigated due to the fact that no one method had proven to be superior.

Shields stated that based on his observations of the results this approach to speaker separation works. The performance of the system was measured by informal listening tests. The best system performance occurred in the recovery of a female speech segment from a male - female combined signal. On the other hand, the poorest performance occurred in the recovery of a male's speech segment from a male - female sum. This can be explained in the following manner. This method of comb filtering allows for a variable number of passbands or "teeth" which depend on the pitch period M. If M is the value of the pitch period at a given time, then, the number of passbands up to the Nyquist frequency (5 kilohertz) will be M/2. In a female voice the pitch period is generally shorter due to many factors such as; size and shape of the vocal tract and articulators. This shorter pitch period would cause fewer passbands to fall within the frequency range of the speech spectrum. With fewer passbands present there is less chance of falsely recovering an unwanted signal. With the roles of the signal reversed, the opposite effect would occur explaining the difficulty in extracting the male's voice from the combination.[19]

During the listening tests several factors were noticed that caused problems in this system: "unitelligible unvoiced segments, short segments of complete distortion of the signal, and breakthrough of the undesired phrase."[20] One cause of the distortion problem was attributed to the rapid changes in the pitch period over a short interval of time. Another cause of the distortion stemmed from the inaccuracies in marking the pitch period which was carried out by hand.

The problem of incomplete or insufficient attenuation of the unwanted speech segment was most bothersome during periods of silence. Shields suggested that if periods of silence could be distinguished from unvoiced segments, then unvoiced segments and silent segments could be treated separately.[21]

This approach for comb filtering was also tried with white noise. Shields originally thought that if the noise were wideband with most of the energy outside the passbands, the signal-to-noise ratio would be improved. This method failed to produce the desired result, and it changed the wideband noise into noise that was harmonic. This distorted the output and produced a signal that had a "reedy" sound.[22]

Using noise that has highly harmonic, this method showed some promising results. The suppression and distortion were much better when the fundamental frequency of the noise was below the average pitch of the speaker. The "pop out" effect could still be observed when the passbands of the comb filter moved into the range of the frequencies where the noise was present.[23]

The remainder of Shields' thesis deals with the automatic extraction of pitch period from a combined waveform set by cepstral analysis.

Since another type of pitch detection will be used in this thesis, this section of Shields' thesis will not be reviewed.

The conclusions that Shields reached can be summarized in the following manner. First, this method will separate the combined speech signals of two speakers. Second, this scheme will reduce the effects of harmonic noise. Third, possible areas of improvement include the treatment of unvoiced segments and periods of silence. The listening tests should be conducted in a more sophisticated manner in order to reveal errors in the system. Fourth, the area of pitch detection could be improved. Also, the choice of coefficients may not be optimal, and a better frequency response might be obtained using another set of coefficients. Finally, Shields suggests the usage of a second comb filter that would use the pitch information of the unwanted speaker to block or reject the unwanted speaker.[24]

2.2 Parsons' System

In a recent paper[25] written by Thomas Parsons, the problem of the automatic separation of the simultaneous speech of two talkers is approached in a different manner.

Parsons points out that the brain requires binaural data in order to separate the combined signal, and in some cases, a signal is received over a channel that does not provide this binaural information. Also, Parsons states that in most schemes, the signal enhancement exploits characteristics of one signal or statistical differences between the signals. The properties or statistics of speech are not understood well enough.[26]

The method that Parsons uses to separate the two talkers capitalizes on the harmonic structure of short segments of speech. In the frequency domain the harmonic structure of the speech appears as peaks, and the procedure involves removing the peaks of the unwanted speaker, and then, taking the inverse transform of the remaining spectrum. This is a rather complicated procedure to implement automatically for several reasons. The frequency peaks from the two speakers will overlap in some areas. The overlap problem is compounded by the pitch variations that occur in natural speech, and this variation causes a frequency modulation (FM) distortion of the peaks. Other problems occur from the event of the pitch tracks crossing. This may cause errors that lead to jumping between speakers.[27]

Parsons uses four rules for detecting an overlap in the peaks, and if any of these criteria are met, then, the process of separating the peaks begins. By using the a priori knowledge of the peak shapes, an estimate of the peak may be subtracted from the combined peaks in order to resolve them.[28]

The next item addressed is the determination of the peak shapes. The largest problem is the effect of the frequency modulation on the peak shape. Parsons states that by assuming the pitch variation is linear over a short segment, the peaks at the harmonics can be approximated by a linear FM ramp. He states that for slow pitch rates, the peak shape is approximately the same as in the case of a constant pitch period. The frequency modulation of the pitch period is shown as a quadrature component with an amplitude proportional to the pitch rate, and the peak has a shape that is the second derivative of the in-phase component.[29] This approximation for the peak shape holds for

frequencies up to 3 kilohertz.

The separation of the two speakers is done by synthesizing the spectrum from a knowledge of the pitch contour and the amplitude and phase of every one of the harmonics. The method of synthesis works better than the method of subtracting the unwanted speaker's harmonics from the combined spectra. Parsons states that if the subtraction methods are used, any errors in parameter estimation will result in incomplete cancellation of the undesired speaker's harmonics.

Parsons states that this system is still in its beginning stages, and that problems of unvoiced speech still have to be solved. Other problems encountered are similar to those mentioned by Shields. These problems include pitch detection, tracking one speaker's pitch contour, and the areas of speech that have pitch rates much faster than in normal speech.

Both methods summarized in this chapter represent possible solutions to the speech enhancement problems, and there are probably other existing methods. The purpose of this chapter was to acquaint the reader with some of the methods that had been proposed, and not an attempt to cover all possible methods for speech enhancement.

2.3 Problem Formulation Introduction

The problem at this point was to decide how the methods for speech enhancement could be improved. The system proposed by Shields was used as a guideline because his work had been the only source of

information on the subject before Parsons proposed his procedures.

The primary concern in this thesis was to explore the methods used for speech enhancement in voiced sections. The unstructured unvoiced speech segments were studied, but because of its lack of structure, the procedures were, for the most part, trial and error. Chapter IV formulates the method that was proposed as an alternative to the comb filter. The method is developed around the structure and time-varying properties of the speech waveform.

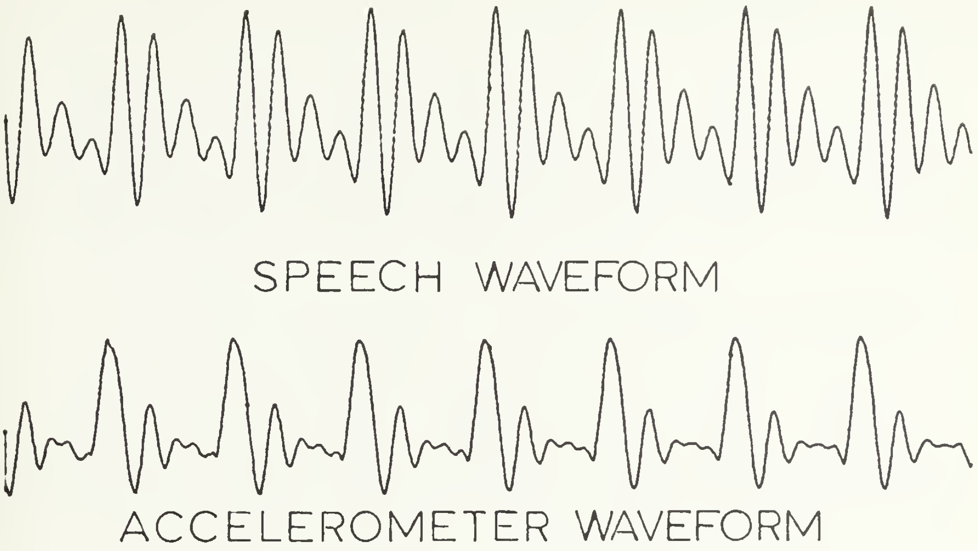
CHAPTER III

PITCH DETECTION PROCEDURES

The need for a simple and accurate algorithm for determining pitch period or fundamental frequency can not be over emphasized. If the separation of a speaker and noise is to be done in some small amount of time by the methods of comb or adaptive filtering, most of the pitch detection algorithms have to be discarded because of their complexity and execution time. The accuracy in determining pitch period is also very important. The filtering techniques that are employed in this thesis use the pitch period information to set the time-varying digital filter. These methods of filtering assume that the pitch period information is exactly known for a given speech waveform, and therefore, errors in pitch detection downgrade system performance. An alternative approach for pitch detection is presented in this chapter.

3.1 Research Performed by Henke

In a recent paper, Henke[30], discusses the features of an accelerometer signal that measures glottal movements may lead to pitch period detection for a speech waveform. The waveforms shown in figure 3-1 immediately show why an accelerometer waveform is more desirable from the standpoint of pitch epoch marking than the corresponding speech waveform.



SPEECH AND ACCELEROMETER WAVEFORMS

Figure 3-1

Henke states that a small accelerometer can be attached on the midline in the suprasternal notch (2 or 3 centimeters below the glottis) to measure the varying pressure waveform. He suggested that the most prominent and stable feature, the "flyback stroke", be used in pitch period detection algorithms. The "flyback stroke", shown in figure 3-2, is a feature of the pressure waveform that is characterized by a rapid change from outward to inward acceleration. This feature occurs immediately following the maximum outward acceleration. The maximum outward acceleration occurs at the instant of closure of the glottis or shortly thereafter. Therefore, the negative going segment of the waveform that crosses the zero line creates a stable point for pitch epoch determination.[31] After these pitch epochs have been marked, it is a simple task to determine the pitch period or fundamental frequency from these markings.

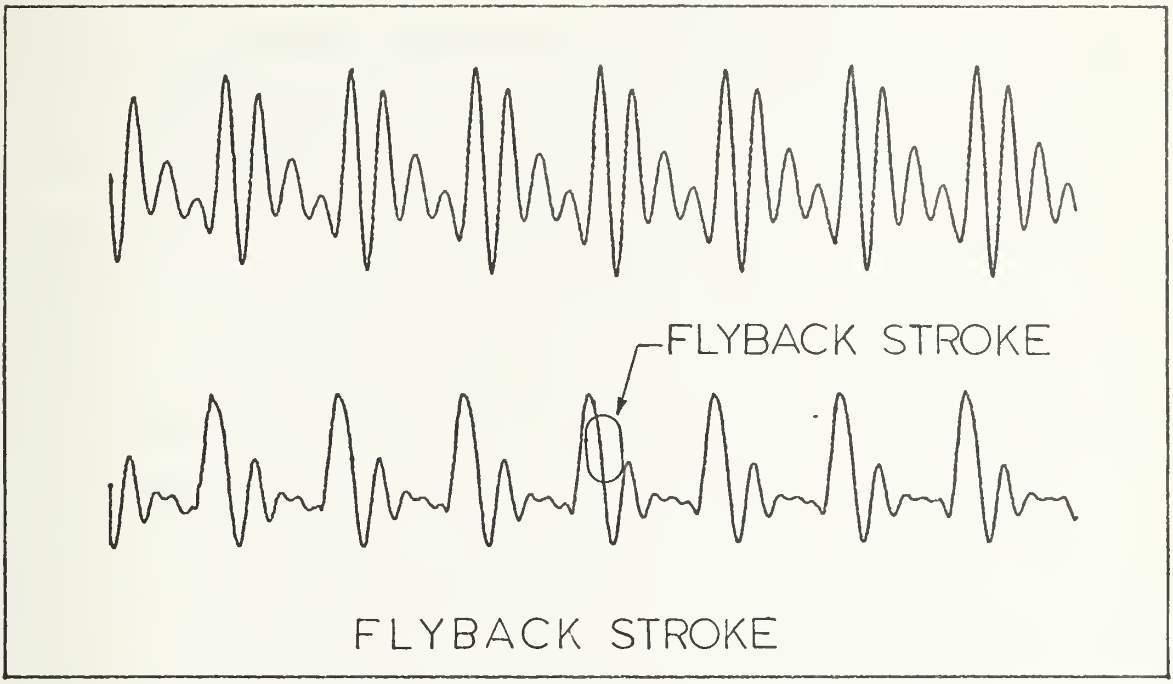


Figure 3-2

3.2 Methods of Pitch Detection Used

Henke's method was used in the work carried out in this thesis. A program was written to process a pressure waveform that was obtained from an accelerometer at the same time as the speech signal was recorded. This was done in a two channel interleaving mode at a sampling rate of 10 kilohertz with the speech signal on one channel and the accelerometer signal on the other.

A program was implemented to find all peaks and zero crossings in a segment of the accelerometer waveform. These peaks and zero crossing locations were stored in two separate arrays, and a threshold was used to determine which peaks were the ones that were associated with the instance of closure of the vocal cords. The first threshold

method consisted of simply setting a constant threshold, by guessing before the processing took place. This threshold was used throughout the program to determine which peaks were the correct ones. The constant threshold approach yielded two types of errors that can be described with figure 3-3. First, the problem of false detection or detecting a peak that was not associated with glottal closure. Second, the false rejection problem had the complementary effect, in that, a true peak was not large enough to be detected by the constant threshold. The constant threshold method yielded about 40 to 50 percent total errors of both types, and the constant threshold was found to be speaker and gain dependent. The errors incurred by this method suggested a second approach for the threshold determination.

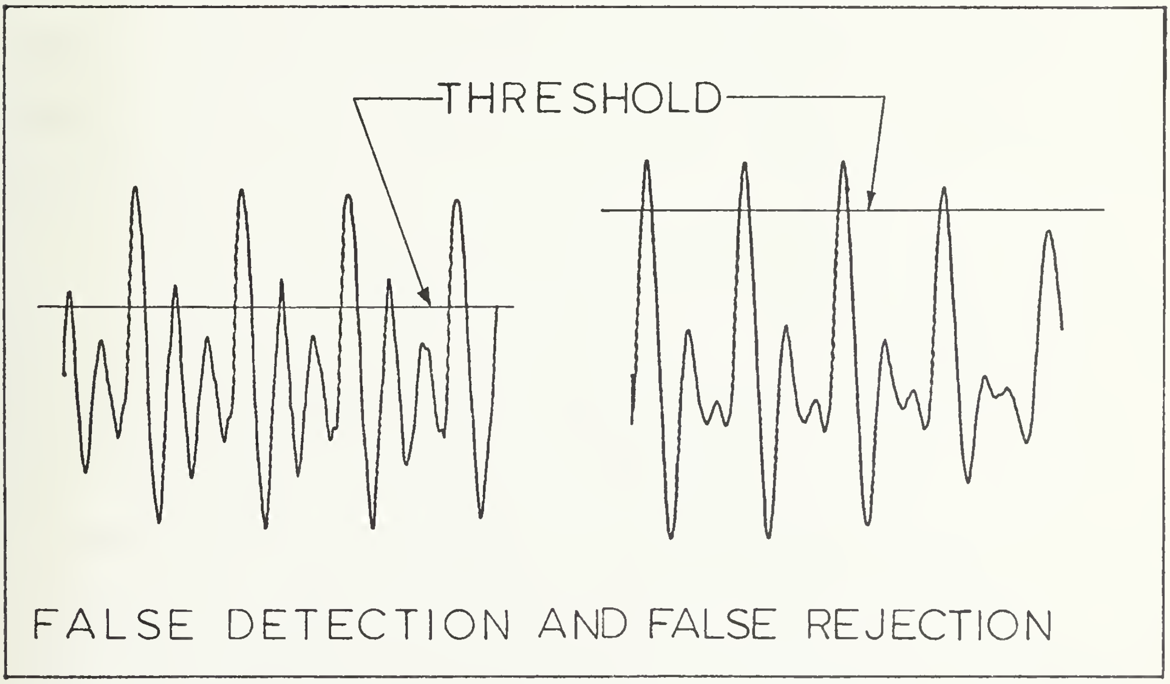


Figure 3-3

The problems in the first approach resulted from the dynamic range of the accelerometer waveform. The second approach consisted of setting a variable threshold. The variable threshold was set by means of an energy measurement of the glottal signal over a short time interval. The energy, E, was measured in the following manner:

$$E = \sum_{n=0}^{N-1} \left[x(n) \right]^2 \tag{3.1}$$

where N was taken to be 25.6 milliseconds.

The range of the values for the glottal peaks was divided into five levels, and after a careful examination of a few waveforms and energy measurements, these ranges were experimentally determined. The results of the variable threshold approach were quite encouraging. The number of errors made was on the order of 20 percent, and this figure could be improved if a more careful statistical study were made to determine the optimal threshold settings for a given energy measurement value.

The idea of the energy measurement also suggested a concept for silence detection. In the thesis done by Shields, he suggests that if silence could be detected, then some other approach could be taken to improve separation. The silence detector was included in the program in order that different schemes of filtering could be tried. A silent area was determined when the value of E was equal to zero. [The silent areas were marked in the table of pitch period values by a minus sign in front of the pitch period value.]

After the peaks were detected, zero crossings in the accelerometer signal immediately following the peaks were marked. The zero crossing was the termination of the "flyback stroke", and as mentioned before, is thought to be one of the more prominent and stable features in the glottal waveform. These zero crossings were marked by using the second channel of the file, since the signal only required one channel.

Since the minimum pitch period is around 2.0 milliseconds, the algorithm included a rule that repositioned the pointer in the waveform by 20 samples after a correct peak was detected. This helped to speed up the algorithm somewhat.

There was another problem that is still associated with this method of pitch determination. The transitions just before voiced and unvoiced sections were not of the same form as during voiced sections or unvoiced sections. Even looking at the section of the waveform that was considered a transition area did not help in the decision of where to put the marks for the pitch period. Each signal had to be marked by hand in these areas, and fortunately, there were not many of them in a waveform.

The outputs of the pitch detection included two separate files. One file consisted of the pitch marks. This file could be added to a speech signal for the filtering of the signal, or it could be added to an accelerometer signal to determine whether the pitch detection algorithm had made any errors. The second output file consisted of pitch period values that were determined from measuring the distances

between the marks in the previous file. The file is referred to as the pitch table, and it is used in the filtering programs.

The methods used in this section were intended to make the pitch detection as easy as possible. The filtering schemes used in the thesis assumed that the pitch information was known exactly. The primary function of the automatic pitch marking program was to eliminate as many periods as possible that would have to be marked by hand. If pitch period detection accuracy had been required in the automatic marking program, more improvements would have been made.

CHAPTER IV

ADAPTIVE FILTERING CONCEPTS

4.1 Development

The idea for the pitch synchronous adaptive filter was developed from the ideas of Siamak Samsam as an alternative to the conventional comb filtering techniques.

The name comb filter probably came about from its frequency response characteristics. The frequency response of a comb filter, shown in figure 4-1 consists of a series of "teeth" or "fingers".

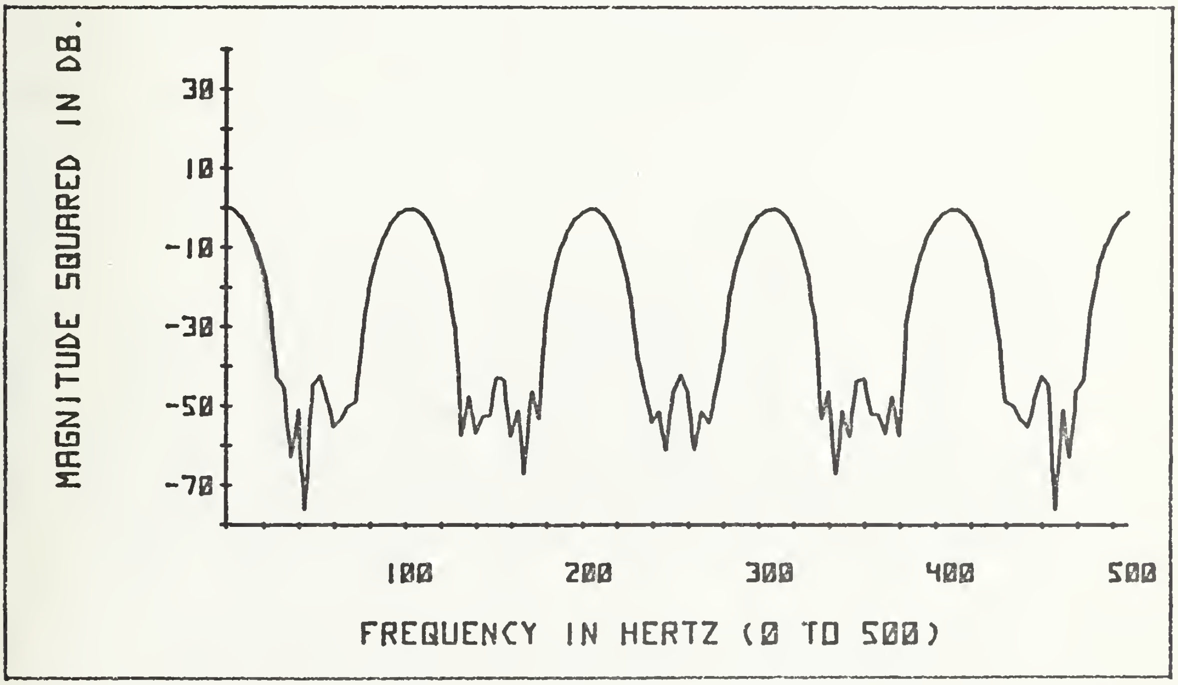


Figure 4-1

The implementation used for the comb filter in figure 4-1 was discussed in section 2.1. The basic idea behind this implementation was to pad the impulse response of a low-pass window function with zeros, using the same number of zeros between the coefficients as the pitch period of the signal at that particular time. The spacing between the coefficients of the impulse response was uniform giving the frequency response in figure 4-1.

In this type of comb filtering implementation the length of the filter was on the order of five to nine times the length of one pitch period, and this is equivalent to a range of filter lengths between 40 and 140 milliseconds. In a normal speaking voice the pitch period or fundamental frequency can change rapidly over several periods depending on intonation or stress, and if the filter length is long, the comb filter may not handle the correct samples by the procedures discussed in section 2.1. The phenomenon discussed above can probably be more easily explained by figure 4-2.

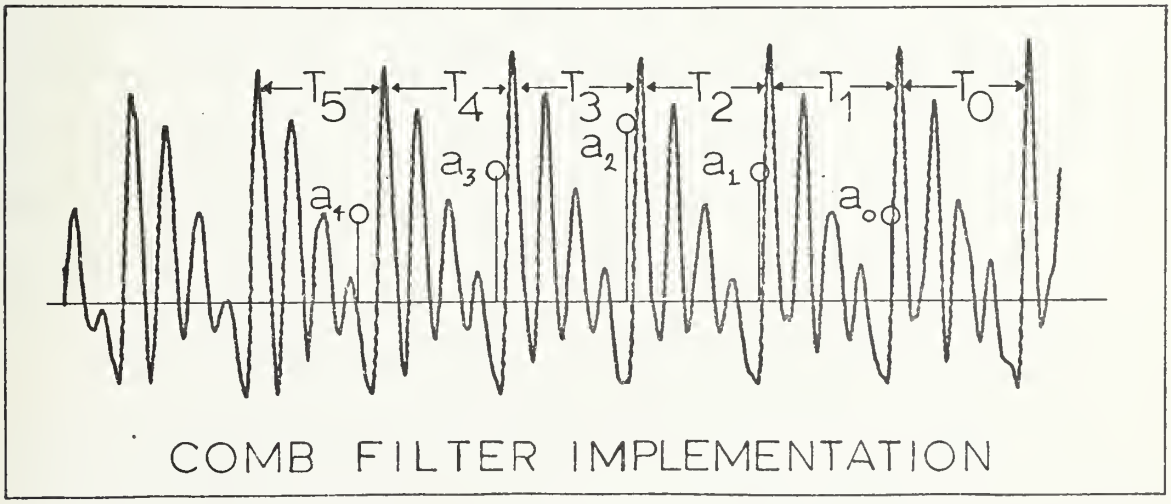


Figure 4-2

Suppose that the signal in figure 4-2 is a speech waveform with pitch periods T_p; where p = 0 to 5. It can be seen that the values of the pitch periods vary over the length of the filter. Suppose that the number of zeros between coefficients has just been changed to correspond to the pitch period T_o, and all spacings between the samples are adjusted so that they correspond to the period T_o. This results in a misplacement of the coefficients in the waveform with respect to the beginning of each pitch period. In the case of a_3, its position in that particular period is much different from the position of coefficients a_o, a_1, and a_2.

Now, suppose that the pitch period T_o, has a smaller error due to the pitch detection program. Since all spacings are uniform, the error is propagated and compounded in the subsequent spacings.

With some insight into what the comb filtering was trying to do, the adaptive methods were formulated. Suppose that the coefficients are rearranged so that the spacings are no longer uniform, but instead, the spacings will be made equal to the pitch period for that particular point in the waveform. Figure 4-3 shows the placement of the coefficients for the adaptive filtering methods. For this method each coefficient is in the same relative position in its particular period.

This method does not really correspond to a comb filter as shown by the frequency response of a sample adaptive filter in figure 4-4. The most substantial thing that can be said about the structure of the frequency response shown in figure 4-4 is that it has no general

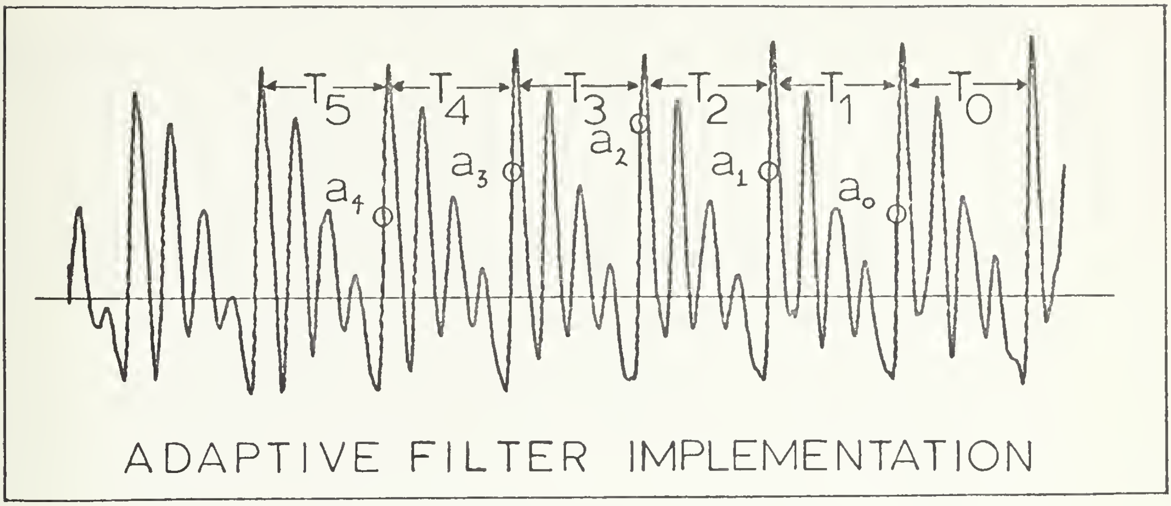


Figure 4-3

structure. The structure of figure 4-4 points out the frequency response of the adaptive filter provides little insight into its capabilities in the speech enhancement problem.

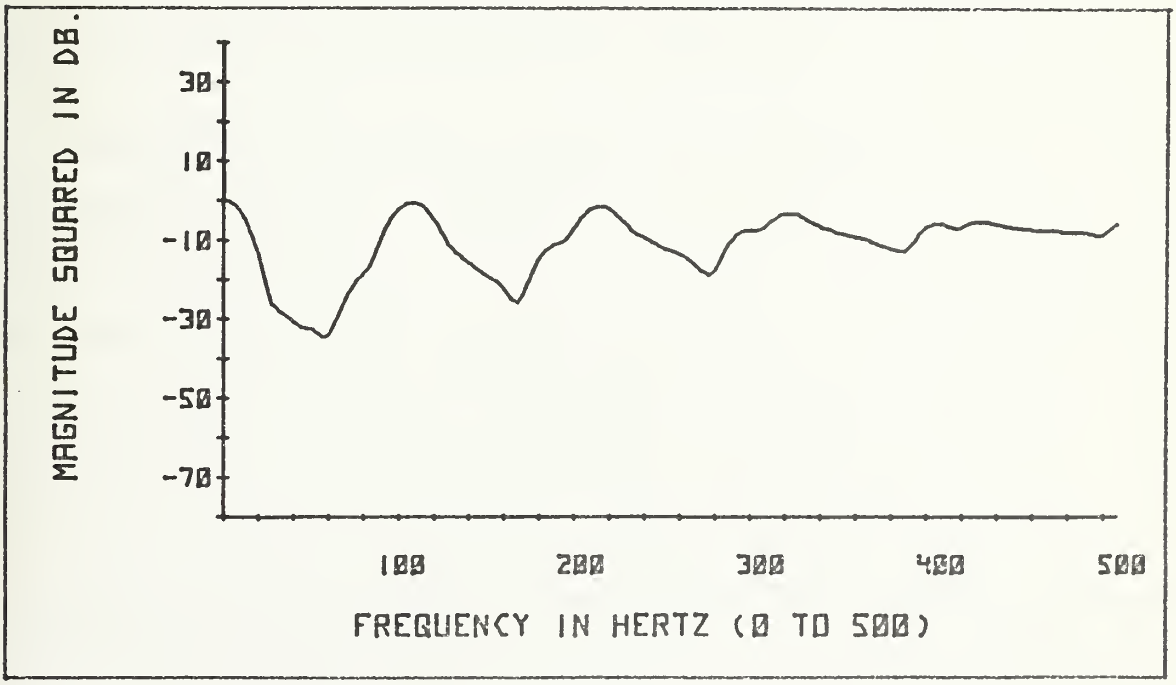


Figure 4-4

The output of this system can be expressed in the following equation:

$$y(n) = \sum_{i=0}^{2K} a_i \; x(n-l_i)$$

where: a_i = the coefficients or weightings

$l_1 = 0$

$l_2 = m_1$

$l_3 = m_1 + m_2$

.

.

$l_i = m_1 + m_2 + \ldots m_{i-1}$

and, m_i = the pitch period of the i^{th} period from the originating point.

$2K + 1$ = the number of coefficients used.

There is another method of expressing the ideas given earlier. This description is aided with the use of figure 4-5.

The method can be thought of in the following manner. First, the waveform is broken into segments according to the pitch epoch or positions where the pitch marks occur. Then, these segments are lined up as shown in figure 4-5. Then, a weighted average is computed point by point as the filter moves in the direction indicated. What this operation intuitively does is to develop an average impulse response based on the previous several periods. This technique works well in conjunction with the assumption that the impulse response of the vocal tract is slowly varying. Another important point to be said for this

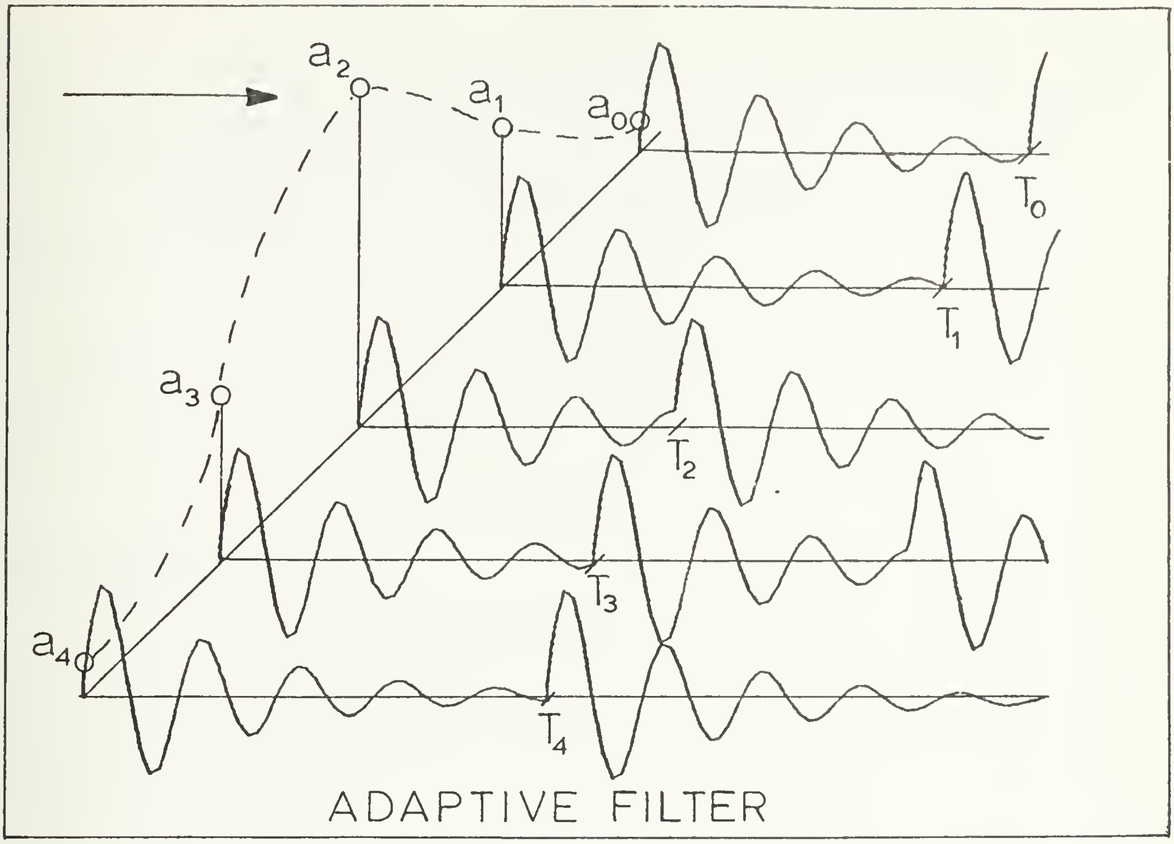


Figure 4-5

technique is the fact that in the separation problem, the weighted samples of the desired speaker will add coherently while the contributions of the undesired speaker will not add coherently. In the comb filtering methods the same statement can not be made as strongly when the pitch is changing. Since the averaging process in the comb filter looks at different relative positions in each period, the reinforcement or coherent addition of the desired speaker is not as pronounced as in the adaptive methods.

It should be pointed out that both methods are identical, and in fact, a comb filter when the pitch period is constant. In other words, the adaptive filter approaches a comb filter when the

pitch period contour is slowly varying and performs a comb filter operation when the pitch period contour is constant. When the pitch contour is fluctuating rapidly, the adaptive filter does not resemble a comb filter in the least.

There is one problem in this method, and this problem with a possible solution is discussed in the next section.

4.2 Overload Problem

A problem in the adaptive filtering method referred to as the "overload problem or phenomenon" can be described with the aid of figure 4-6.

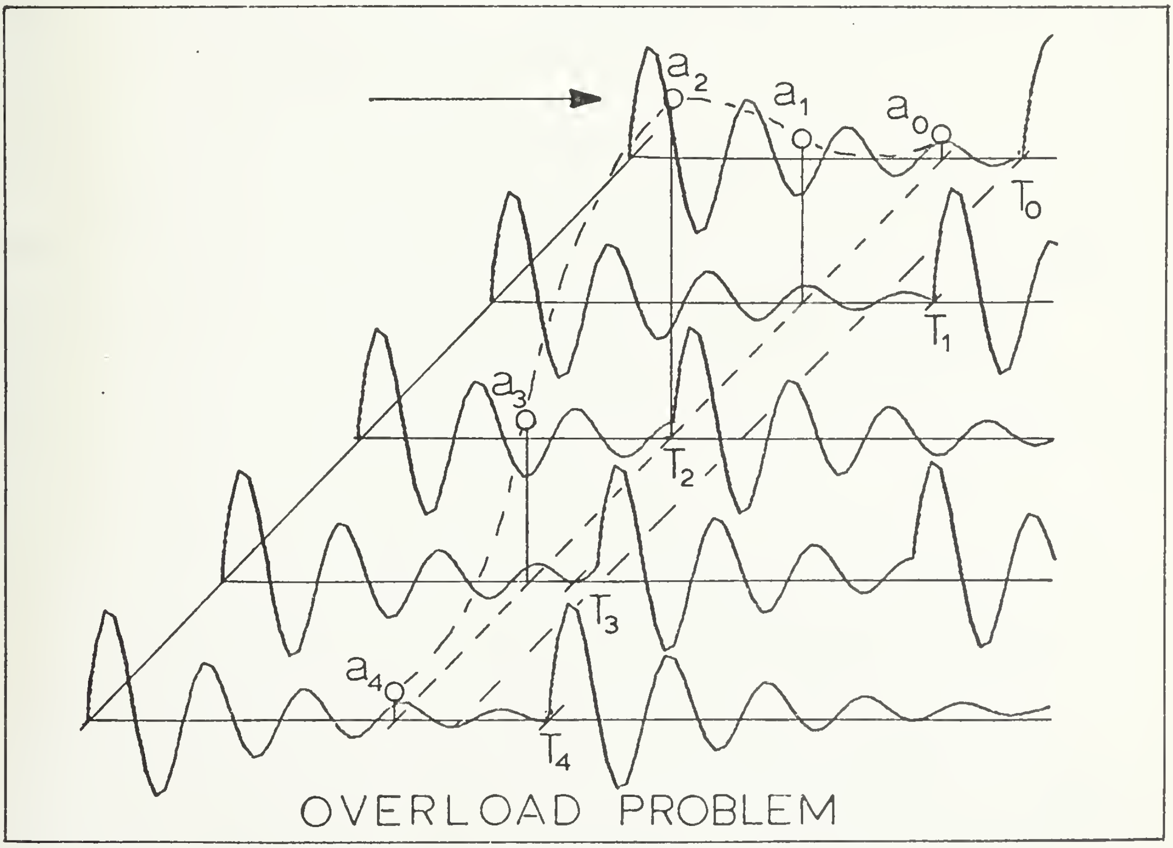


Figure 4-6

Suppose that there is a rapid increase in the pitch period in a relatively short interval of time. This phenomenon is not rare in normal speech. For some speakers a distinguishable feature or characteristic is the short pitch period that occurs at the onset of voiced speech. This short period can be represented by the segment in figure 4-6 labeled as T_2. As the filtering moves to the right computing a point by point average, the method runs into problems at the point T_2. Up to the point T_2 the filtering has occurred in the correct manner. After T_2 the coefficient a_2 begins to use samples in segment two for the averaging. This feature disagrees with the concepts of the adaptive filtering methods because the coefficients now are not in the same relative positions in their respective segments. This phenomenon has been named the overload problem and causes an undesirable result in the output waveform. The overload phenomemon will be studied more in the next chapter.

There are several proposals for solutions to the problem of overloading. First, the easiest solution from all viewpoints is to ignore the problem and let the overloading occur. If the pitch period is slowly varying, the problem will be minor and the coefficients will not be far away from the same relative position in their respective periods. Second, the adaptive filter could be "turned off" at T_2 and pass the input signal completely for the time interval between T_2 and T_o . This idea originated from the fact that the impulse response of the vocal tract would be of sufficiently low amplitude in the tails of the impulse response so that the filtering operation would not have much effect in this area. The final solution proposal, and the one

that was used in this thesis was one that padded the short segments in the filter's length with zeros in order to make the length of the short segments equal to the one in the front of the filter.

Figure 4-7 explains this procedure. The shortest period, segment 3, will have zeros added in the area indicated.

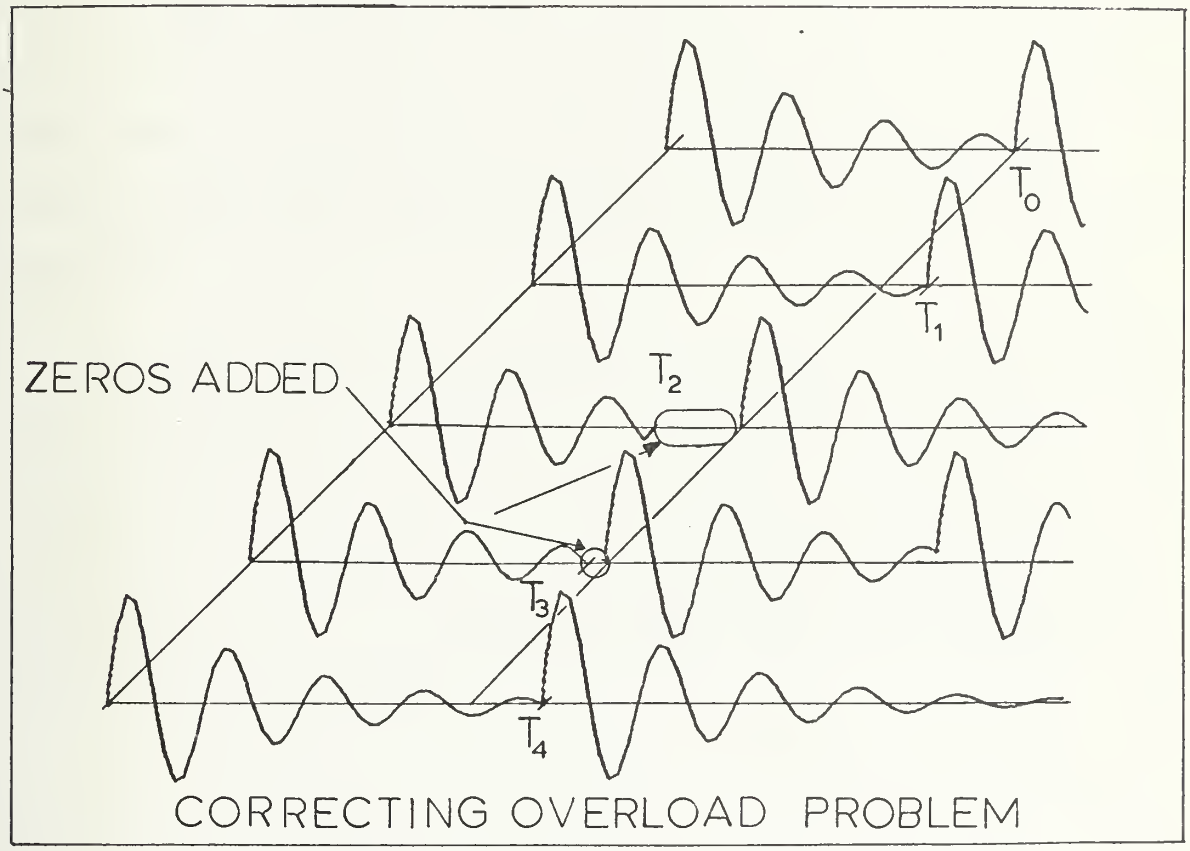


Figure 4-7

When the filter reaches the end of segment 4, the same problem occurs and zeros are again added to pad out this segment. In effect this is the same procedure as "turning off" the individual coefficients associated with the segments that have been padded with zeros. In figure

4-7 this would involve setting a_2 equal to zero after T_2 is reached and a_3 equal to zero after T_3 is reached.

There is one other factor to be considered in this method. The gain associated with the filter has been fixed to be a unity gain system by:

$$\sum_{k=0}^{2K} a_k = 1 \qquad (4.2)$$

When a particular coefficient is set to zero or "turned off", the remaining coefficients must be rescaled so that the output is not attenuated.

The third procedure for handling the overload problem makes sense from an intuitive standpoint. It says that if a coefficient can not be placed in the same relative position in the segment or period due to the shortness of that period, then, the contribution of that coefficient will only provide erroneous results, and therefore, it should not be considered.

4.3 Rules

As the filtering algorithms become more complicated and complex, the list of rules for handling various situations also grows. Some of the more important rules used will be discussed in this section.

The discussion thus far has dealt only with the procedures for handling voiced speech. Normal speech waveforms contain unvoiced sections also, and these sections must also be handled. There are

three basic areas or situations involved in the processing of unvoiced speech sections. These situations are:

1. The detection of an unvoiced section, and transition from voiced speech procedures to unvoiced speech procedures.
2. The procedures that are used when the filter is entirely in the unvoiced section.
3. The detection of a voiced section, and transition from unvoiced speech procedures to voiced speech procedures.

The first situation mentioned above was handled by using the last voiced pitch period as shown in figure 4-8.

In figure 4-8 the speech waveform has been omitted for clarity, but the sections are marked as voiced and unvoiced. The values labeled m_i are equal to the pitch period of that particular period. In figure 4-8 (a) the filter position is shown just prior to an unvoiced area detection. Figure 4-8 (b) shows the spacing after the first coefficient has entered the unvoiced area. In this figure two of the spacings have a spacing of m_1 while the other two spacings are related to their particular periods. In figure 4-8 (c) another spacing change is shown so that three spacings are equal to m_1. This procedure is repeated until the entire filter lies in the unvoiced area.

Now the second situation has been entered, and another scheme is begun. Since the method of attenuating the input in an unvoiced section worked satisfactorily in the system proposed by Shields, it was used in this system also. In effect, the filtering was terminated,

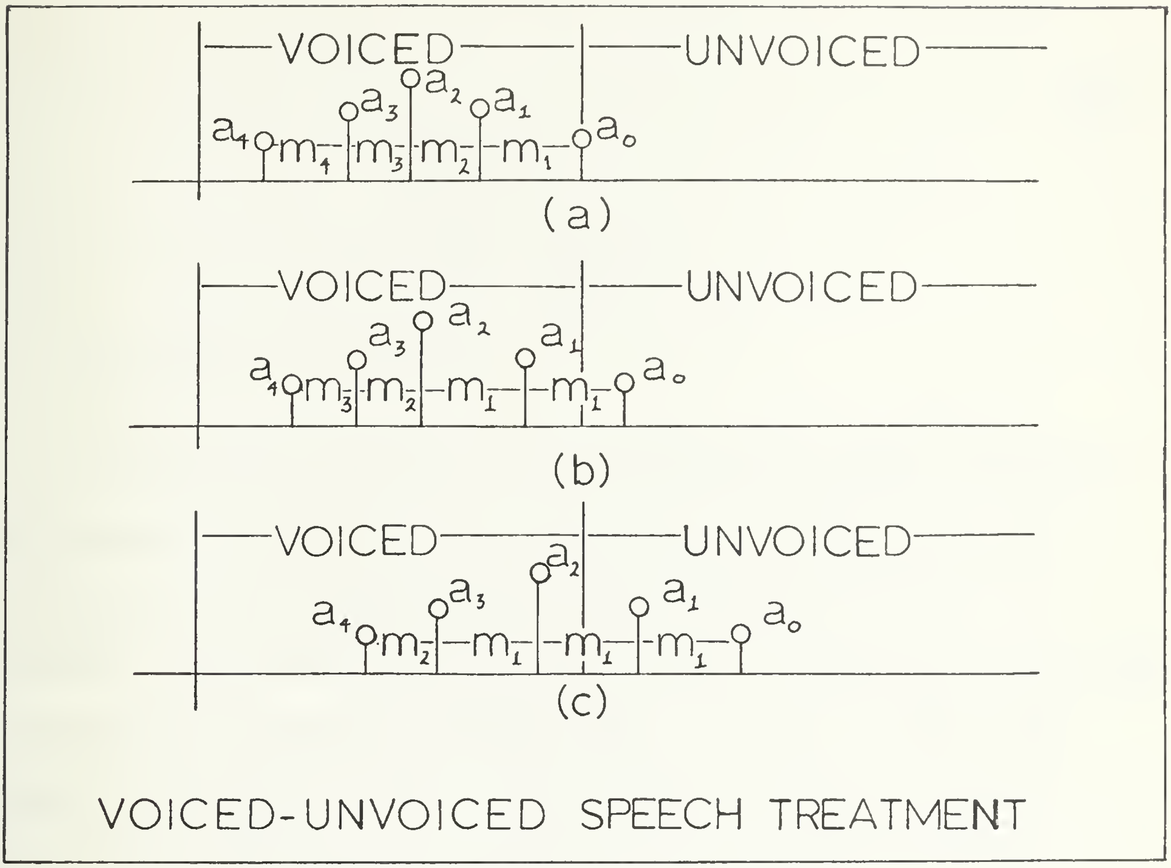


Figure 4-8

and the input waveform was multiplied by a constant less than one in order to produce the output waveform.

In the third situation the procedure was basically the inverse of the method used in the first situation. When a voiced section was detected, the filter was initialized to a constant spacing using the first period value. As the filter moved further into the voiced sections, the spacings were changed to conform to the pitch periods.

In summarizing this chapter the most prominent feature of the adaptive filtering approach is its conformity with the speech waveform. The performance of this system will be evaluated in later chapters.

CHAPTER V

TEST SIGNAL SECTION

5.1 Development of Test Signals

It was decided that before any new methods for speech enhancement were developed, a method of testing the various schemes should first be undertaken. Even though psychological listening tests were to be conducted in the future, there were just too many variables in the system to be handled in a complex listening test. In order to assign some types of performance indicies, and in order to fix some of the variables, a test input signal was generated and stored for future use.

The purpose of the test input signal was to alleviate the uncertainty of characteristics of the speech waveform, such as, the pitch period and the impulse response. When the pitch period and the impulse response of a voiced section of speech are known, then, some types of input and output waveform comparisons can be made in order to determine optimal filter properties.

A test input signal was formulated in the following manner: First, it was decided that enough comparison data could be obtained from a relatively short segment of the input signal. This feature would provide results that were accurate, and in addition, they could be obtained quickly. Second, it was decided that the impulse response

of the generation filter would be time-invariant over the duration of the signal. This assumption was chosen in order to make the comparison study easier, and it conforms to the properties of speech, in that, the impulse response of the vocal cords over short time segments can be considered constant. Third, it was decided that the pitch period would vary sinusoidally about some mean pitch period.

The input signal formulation can be described by the following diagrams and equations:

Let $x(n)$ = the test input signal.

$v(n)$ = the impulse response of the generating filter.

$w(n)$ = the nonuniformly spaced impulse train corresponding to the pitch pulses or the excitation.

Then, $x(n) = v(n) * w(n)$ or (5.1)

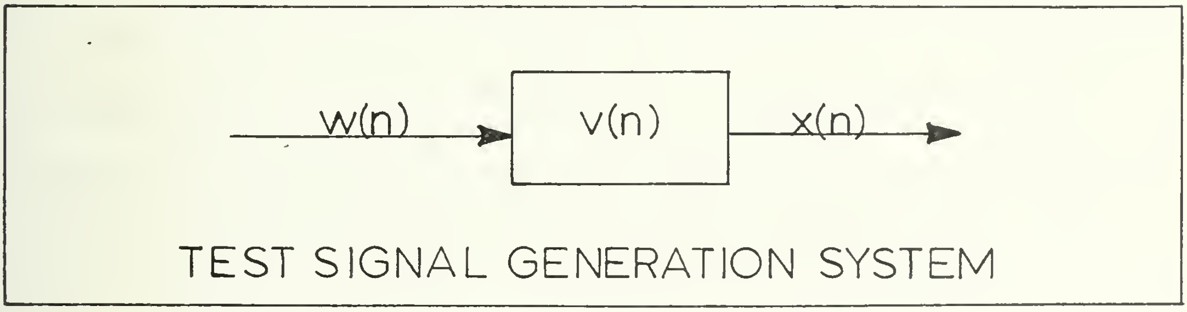


Figure 5-1

The impulse response of the generating filter, $v(n)$, was chosen to be a damped sinusoid of the following form:

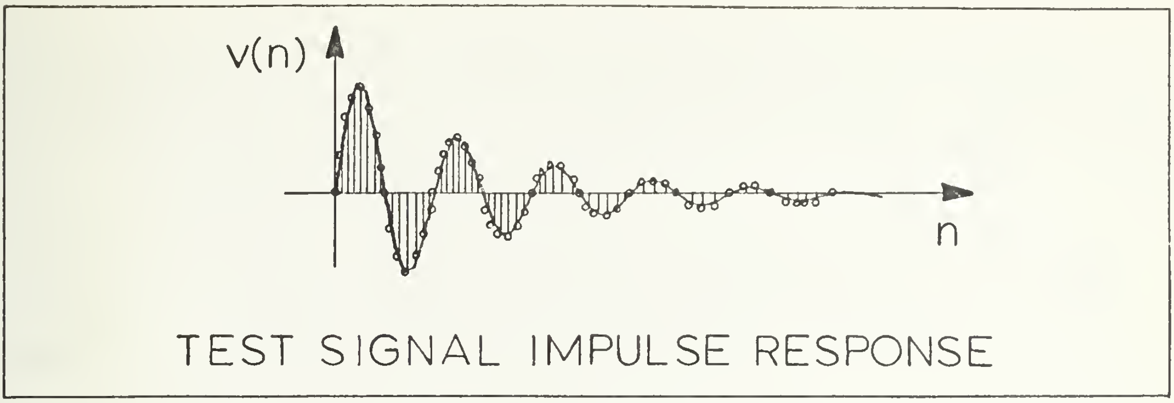


Figure 5-2

where: $v(n) = e^{-n/\alpha} \sin 2\pi\omega n \qquad 0 \leq n \leq L-1$ (5.2)

The model seems to be a very good model for the impulse response of the vocal cords. The parameters α and ω were variable and were chosen so that the input signal that was generated would be a good model of speech.

The length of the impulse response was chosen to be $2T_o$, where T_o is the value of the mean pitch period. It was felt that in order to model speech as closely as possible, the length of the impulse response of v(n) should be longer than the pitch periods encountered. This would cause some overlap of the impulse response, v(n), in the input waveform.

The nonuniformly spaced impulse train, w(n), was formulated in the following manner: The spacing between the impulses, (the pitch period) was chosen to vary sinusoidally about some mean pitch period T_o . The figure 5-3 shows the procedure for this development. It can be seen that this nonuniformly spaced impulse train is periodic on the interval $\left[0, N\right]$.

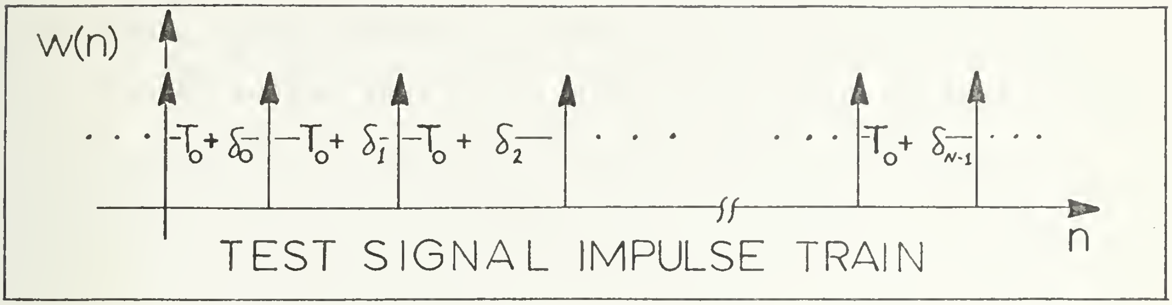


Figure 5-3

where: $\delta_i = \delta \sin \frac{2\pi i}{N}$ for $i = 0$ to $N-1$ (5.3)

Now with these two components the input signal x(n) was generated. This gave a waveform whose impulse response, and pitch contour were exactly known, and the various methods of speech enhancement could be tried on this signal.

5.2 Processing of Test Signals

With the test signals that have been described in the previous section, several processing methods may now be more closely examined. The filtering schemes used in this section were designed so that they resembled the actual filtering systems in all respects except for the length of the input waveforms used. Since the test input waveform was much shorter than the actual speech waveforms, it could be completely stored in core memory, and the amount of time needed for the signal processing was short.

There were three filtering systems implemented for the test signal input waveform. These systems included the system proposed by Shields,

the adaptive filtering systems proposed in Chapter IV with the overload problem, and the adaptive filtering system without the overload problem. For the remainder of the thesis the adaptive system with the overload problem will be referred to as the adaptive overload system. The adaptive system that compensates for this problem will be referred to simply as the adaptive system.

Another problem that received much thought was the problem of comparing these system in some manner to determine the good and bad points of both systems. Several tests were decided upon that were related to the problems of speech enhancement. In the thesis done by Shields, the tradeoffs between desired speaker distortion and undesired speaker separation were described. These tradeoffs were also discussed in Chapter II. It was decided that two separate tests could be conceived for this area.

The first test will be referred to as the input pass test. Figure 5-4 describes the manner in which this was implemented. By using the enhancement system in this manner, the output waveforms should exactly resemble the input waveform, or in other words, the system should act as an identity system. By examining the inputs and the outputs of the systems, the amount of desired speaker distortion could be determined for various numbers of coefficients or windows used.

The second test, described by figure 5-5, will be referred to as the input reject test. In this type of filtering the pitch information is provided from another input waveform that is totally uncorrelated with the test input signal. The idea behind this type of processing

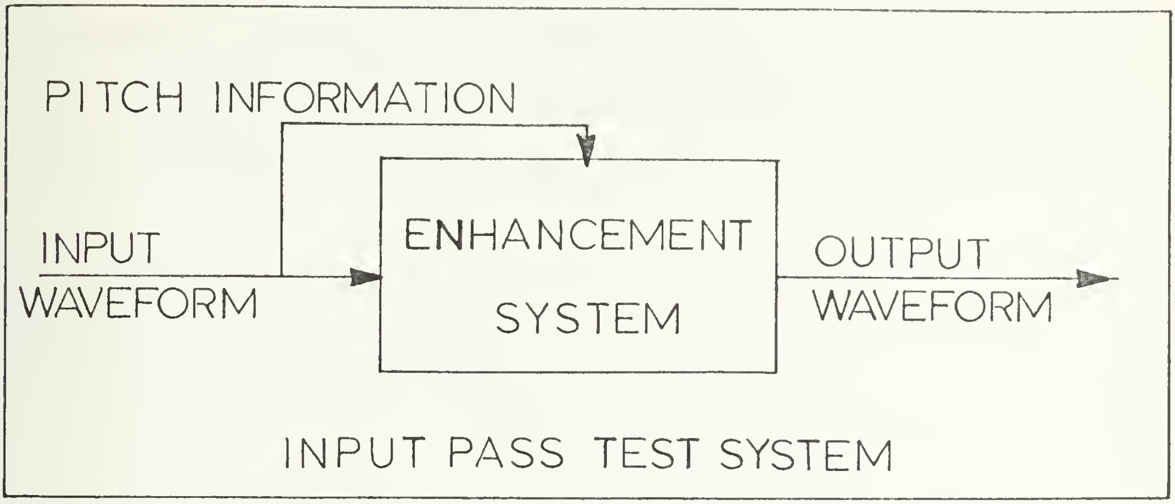


Figure 5-4

is to examine the amount of attenuation introduced into the output waveform. Ideally, the output would be zero, because the enhancement system would be passing another waveform with a different pitch contour. This test was designed in order to measure the amount of speaker separation that resulted from the various systems.

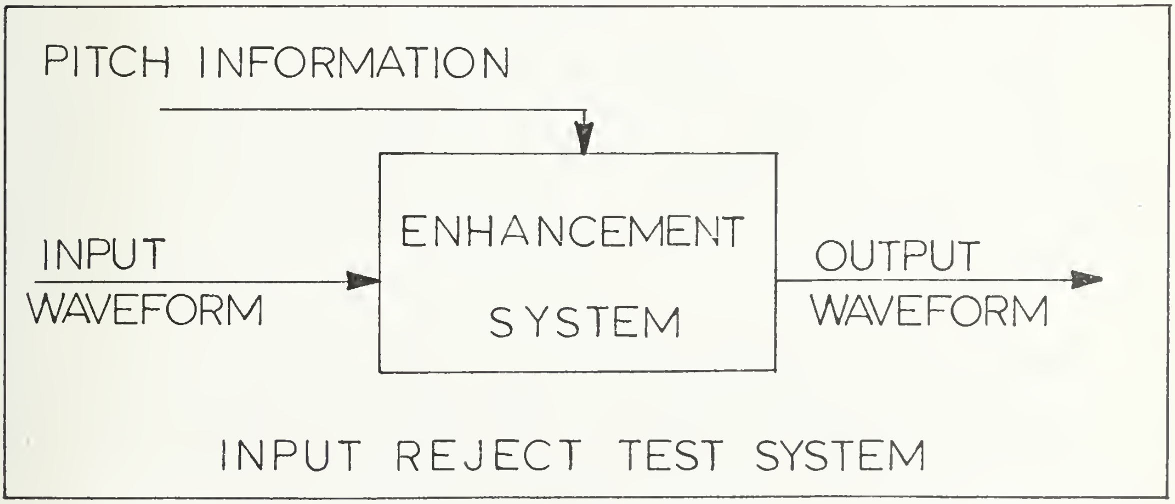


Figure 5-5

Another test was devised in order to examine the capability of each system when white noise was added to the input signal. Figure 5-6 shows the implementation of this system.

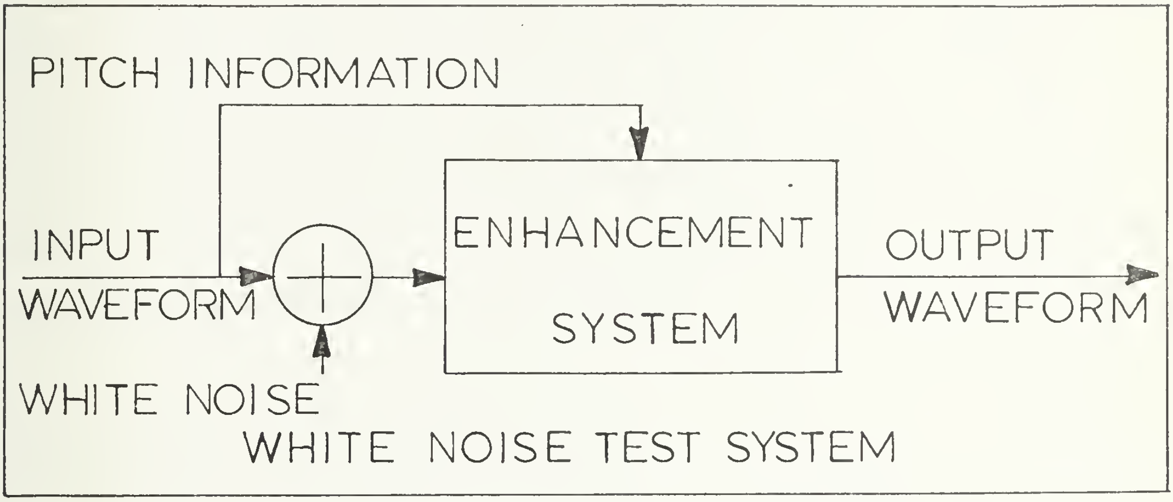


Figure 5-6

Shields stated that the system proposed in his work seemed to turn white noise into a highly harmonic noise in the output waveform that was very annoying.[32] This test was designed to determine if the adaptive filter had a problem with white noise, and if it did, the systems could be compared to determine which one did the best in these circumstances.

Finally, a test could be performed on the sum of two waveforms that would duplicate the problems of the overall speech enhancement system. It should be pointed out that this test was merely a combination of tests one and two, and since the systems were linear, superposition would have given the same results.

In all cases it was assumed that the entire test signal was periodic with a period of twenty pitch periods, and the processing only concerned itself with one period of the overall waveform.

It should be pointed out that the test signals used in these tests were not actual speech waveforms. The input waveform was modeled to closely resemble the speech waveform during voiced sections. There were no comparison tests made on test signals modeled as unvoiced speech. In the next section the results of these tests for the different systems implemented will be discussed.

5.3 Results of Test Signal Processing

In order that the results of the test signal processing be presented as clearly as possible, many figures will be employed in this section. There are three basic types of presentation used in this section. First, a time domain presentation is used to show how the various systems process the input waveforms, and from these figures, the input and output waveforms can be compared visually. Second, a frequency domain presentation is used. The test signal was formulated to simulate a signal that had been sampled at 10 kilohertz. A short time spectrum was computed by resampling the input waveform at 5 kilohertz, multiplying by a Hamming Window, and then, the spectrum was computed using a Fast Fourier Transform (FFT) program. The bandwidth of the test signal was well below the Nyquist Frequency for the new sampling rate. The frequency domain presentation displays logarithmically the square of the magnitude of the Fourier Transform of the

signals, and this was computed to eliminate the effects of any phase errors that were introduced by the filtering. Finally, an error function is shown in some cases, and this error function can be expressed in the following equation:

$$E = 20\log\{|X(e^{j\omega})|^2\} - 20\log\{|Y(e^{j\omega})|^2\} \qquad (5.4)$$

where: $X(e^{j\omega})$ is the Fourier Transform of the input signal

$Y(e^{j\omega})$ is the Fourier Transform of the output signal.

Notice that the error function defined above can be thought of in another manner. Consider the linear system shown in figure 5-7 below:

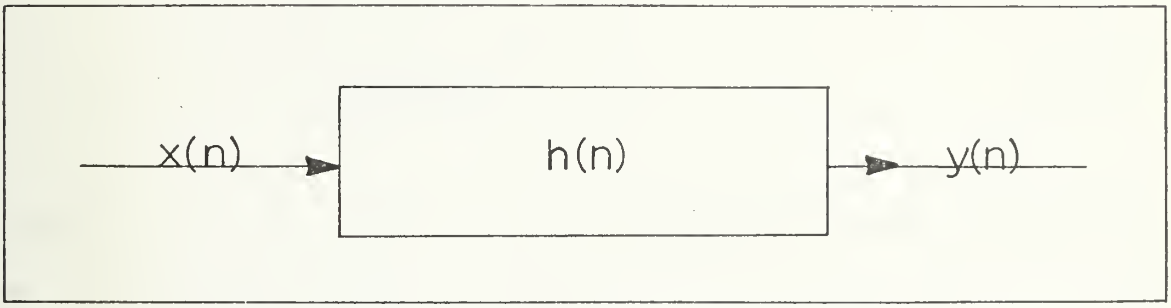


Figure 5-7

Clearly,

$$y(n) = x(n) * h(n) \qquad (5.5)$$

Expressing equation (5.5) in the frequency domain:

$$Y(e^{j\omega}) = X(e^{j\omega})\ H(e^{j\omega}) \qquad (5.6)$$

Now, $$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(eJ^{\omega})} \qquad (5.7)$$

Taking the magnitude and squaring both sides gives:

$$\left| H(e^{j\omega}) \right|^2 = \left| \frac{Y(e^{j\omega})}{X(e^{j\omega})} \right|^2 = \frac{|Y(e^{j\omega})|^2}{|X(e^{j\omega})|^2} \qquad (5.8)$$

Now, taking the logarithm of both sides gives:

$$\log|H(e^{j\omega})|^2 = \log\left[\frac{|Y(e^{j\omega})|^2}{|X(e^{j\omega})|^2}\right] =$$

$$\log\{|Y(e^{j\omega})|^2\} - \log\{|X(e^{j\omega})|^2\} \qquad (5.9)$$

Notice that this is almost the same form as the error function. $H(e^{j\omega})$ is often referred to as the system transfer function, and the error function may be represented as:

$$E = \log\left[\frac{1}{|H(e^{j\omega})|^2}\right] \qquad (5.10)$$

Therefore, the error function could be thought of as being related to the inverse system transfer function.

Before proceeding the following discussion will be covered in order to show how the error function will be used to calculate system performance. Consider the system proposed in figure 5-7. Suppose that the following result was desired:

$$y(n) = a \;\; x(n) \qquad (5.11)$$

Expressing this in the frequency domain and because the system is linear:

$$Y(e^{j\omega}) = a \;\; X(e^{j\omega}) \qquad (5.12)$$

Now, $$\frac{X(e^{j\omega})}{Y(e^{j\omega})} = \frac{1}{a} \tag{5.13}$$

$$\log\left[\frac{|X(e^{j\omega})|^2}{|Y(e^{j\omega})|^2}\right] = \log\left[\frac{1}{|a|^2}\right] \tag{5.14}$$

$$E = \log\{|X(e^{j\omega})|^2\} - \log\{|Y(e^{j\omega})|^2\} = \log\left[\frac{1}{|a|^2}\right] \tag{5.15}$$

$$E = \log(1) - 2\log\{|a|\} = -2\log\{|a|\} \tag{5.16}$$

The above results will be used in later parts of this section for comparative analysis between the systems implemented.

The first test described in section 5.2 was named the input pass test. This test was performed with the three basic systems developed: Shields', the adaptive overload, and the adaptive. For all systems five coefficients were used, and these coefficients were taken from the Hamming Window function.

Figure 5-8 shows the time domain input and output waveforms for Shields' system. Several things can be noticed in the output waveform. First, there is a definite amplitude modulation throughout the waveform. Second, there are some areas where an "overload bump" occurs in the output. Figure 5-9 shows the same presentation for the adaptive overload system. It can be noticed that the amplitude modulation does not exist, but there are still some problems with overloading. In figure 5-10, the adaptive system results are shown: input and output waveforms are essentially identical.

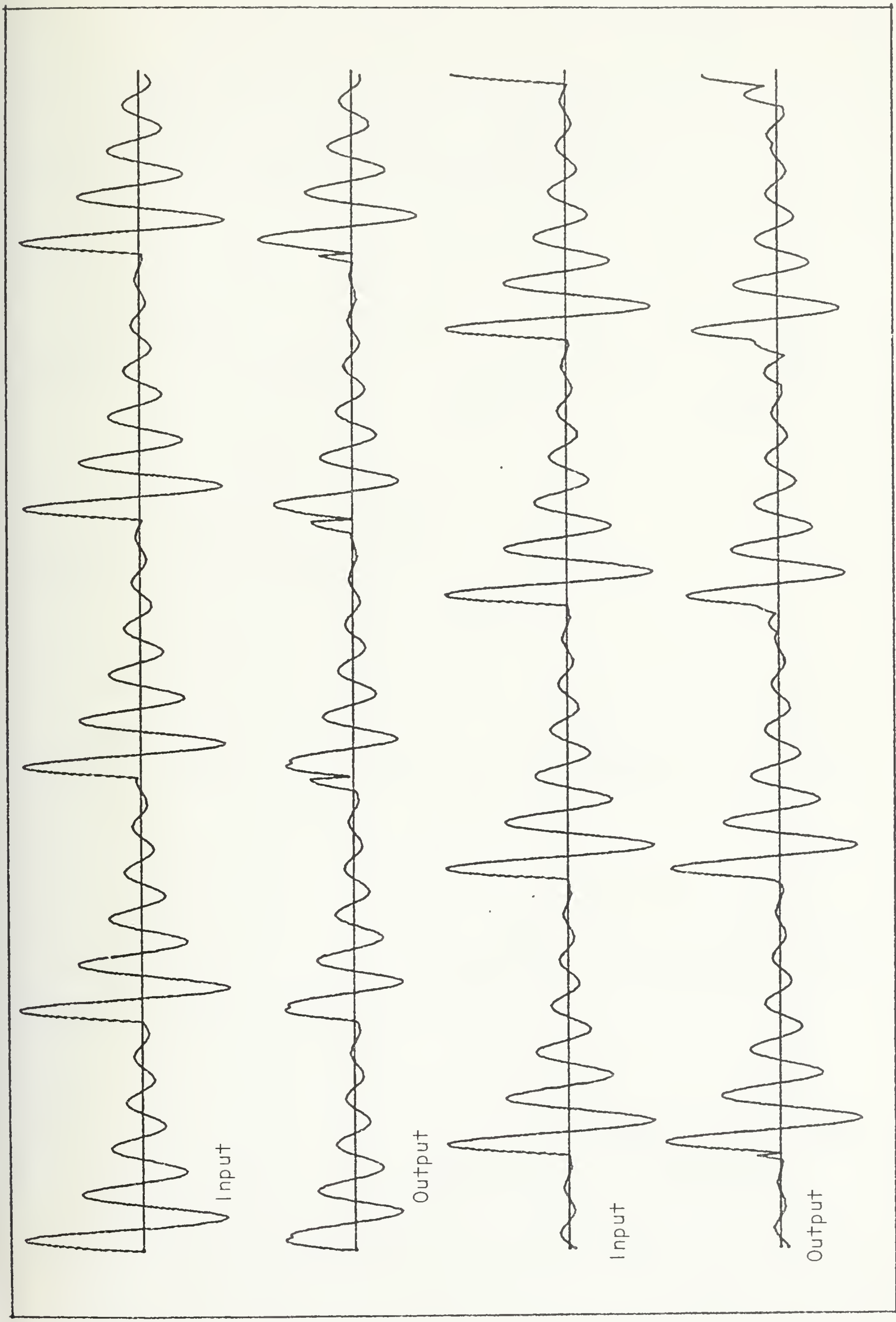


Figure 5-8 Input and Output Waveforms - Shield's System; Input Pass Test

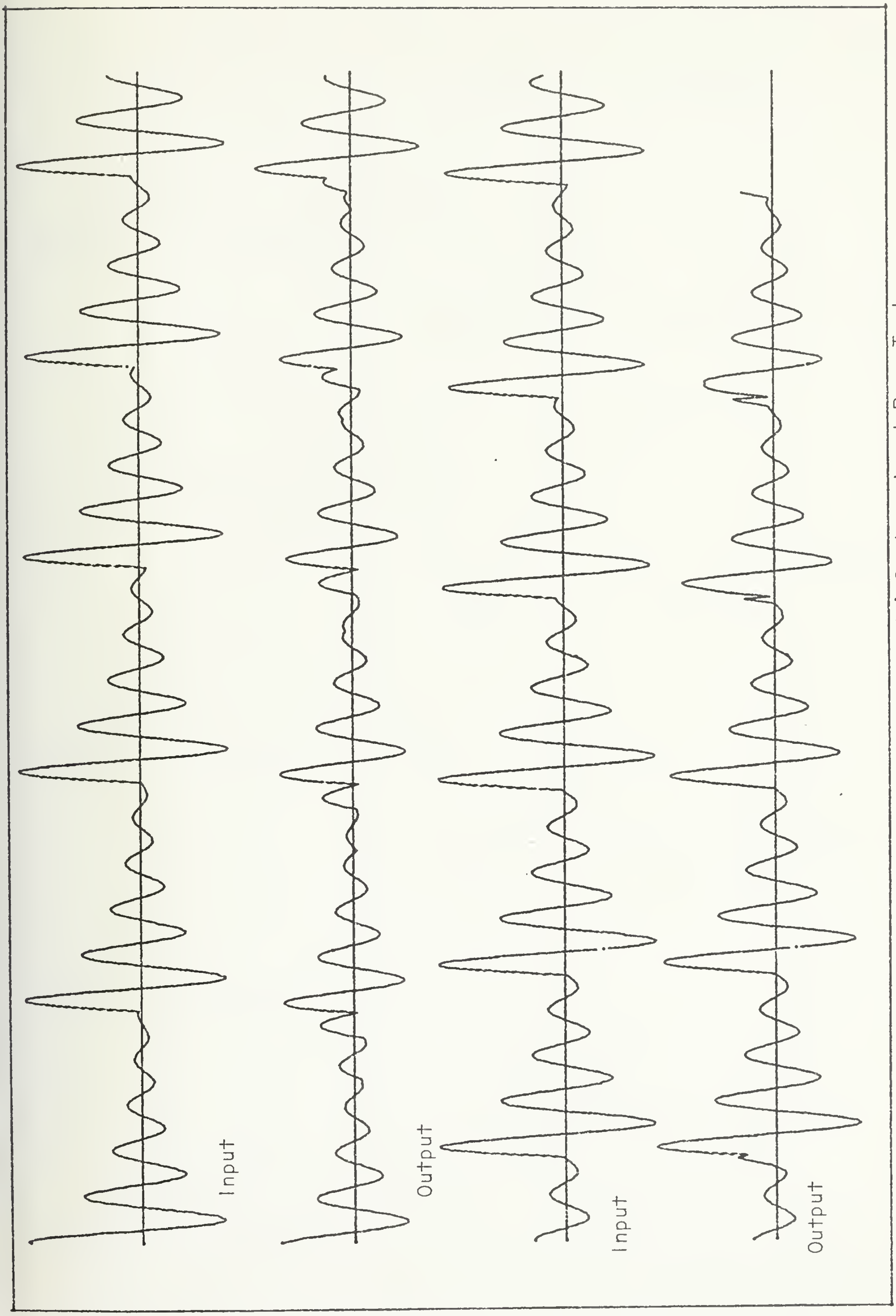


Figure 5-8 (con't.) Input and Output Waveforms - Shields' System; Input Pass Test

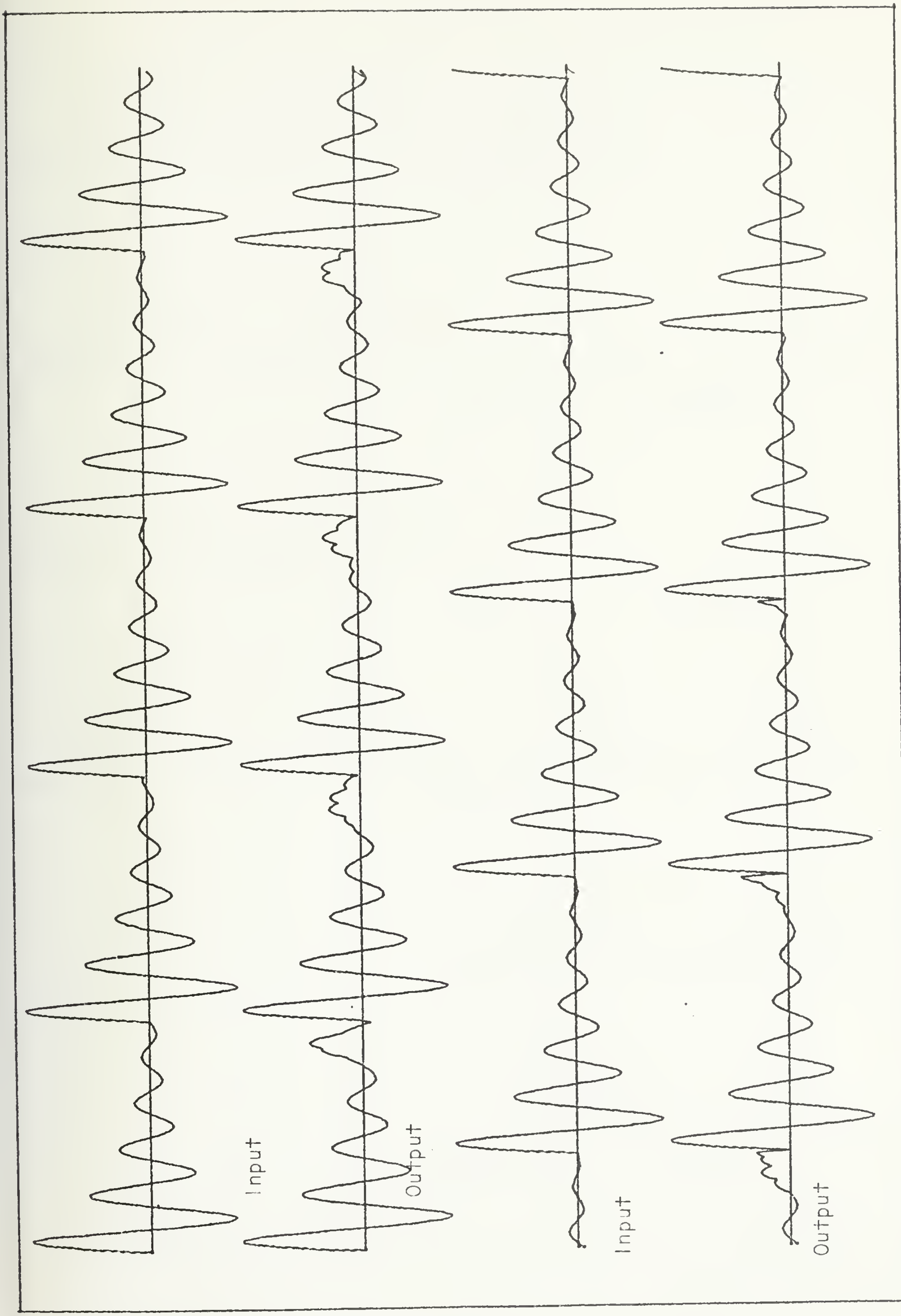


Figure 5-9 Input and Output Waveforms - Adaptive Overload System; Input Pass Test

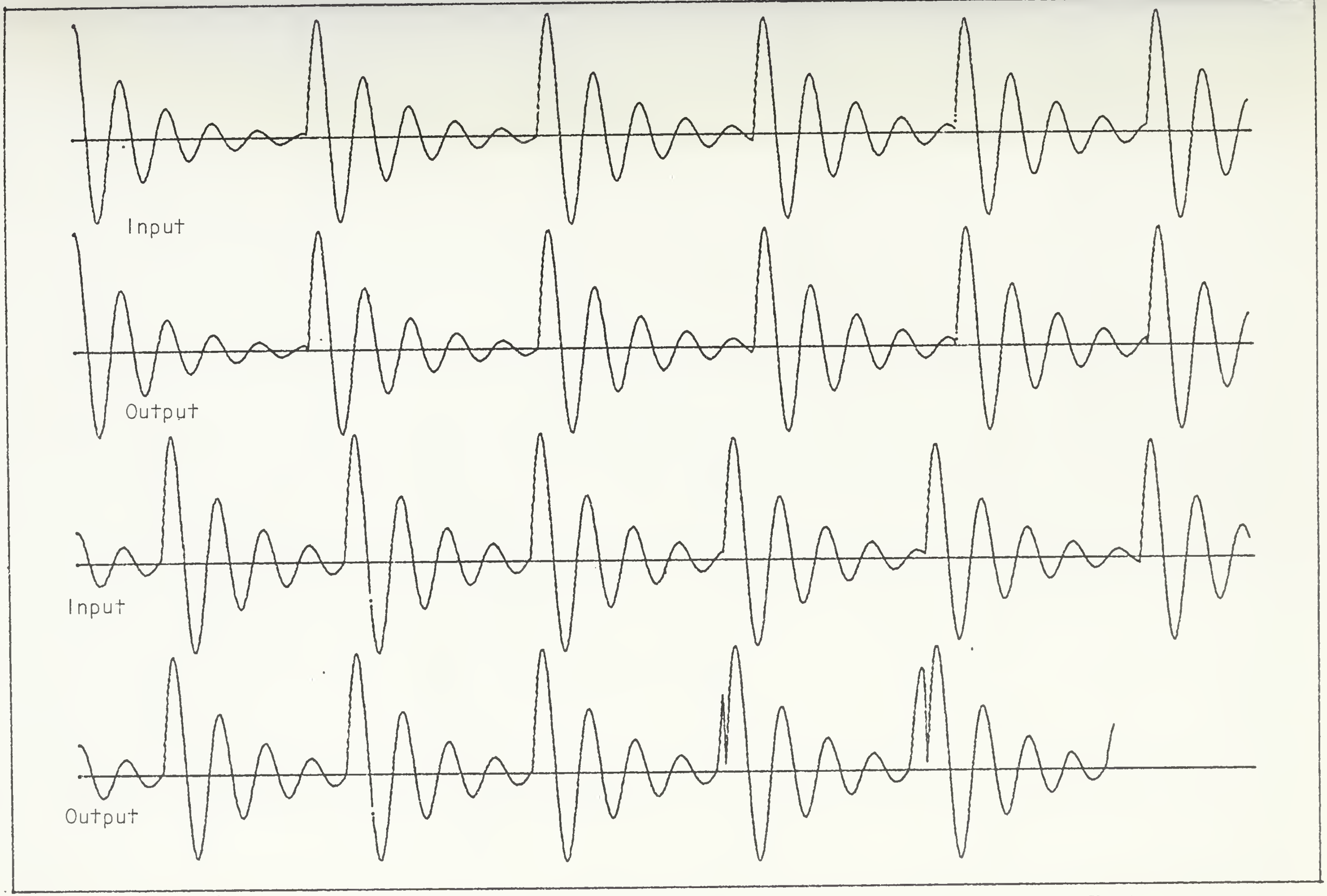


Figure 5-9 (con't.) Input and Output Waveforms - Adaptive Overload System; Input Pass Test

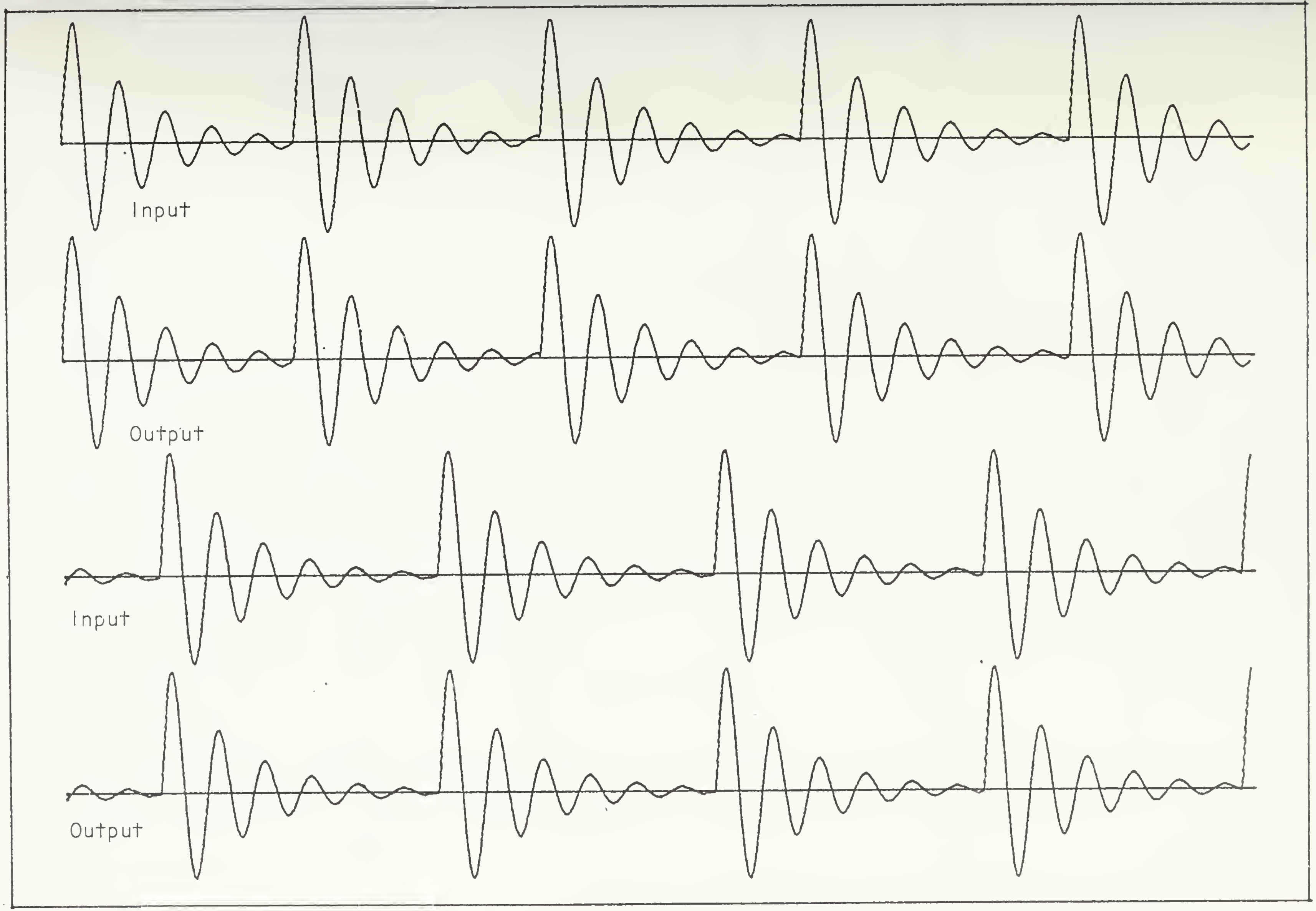


Figure 5-10 Input and Output Waveforms - Adaptive System; Input Pass Test

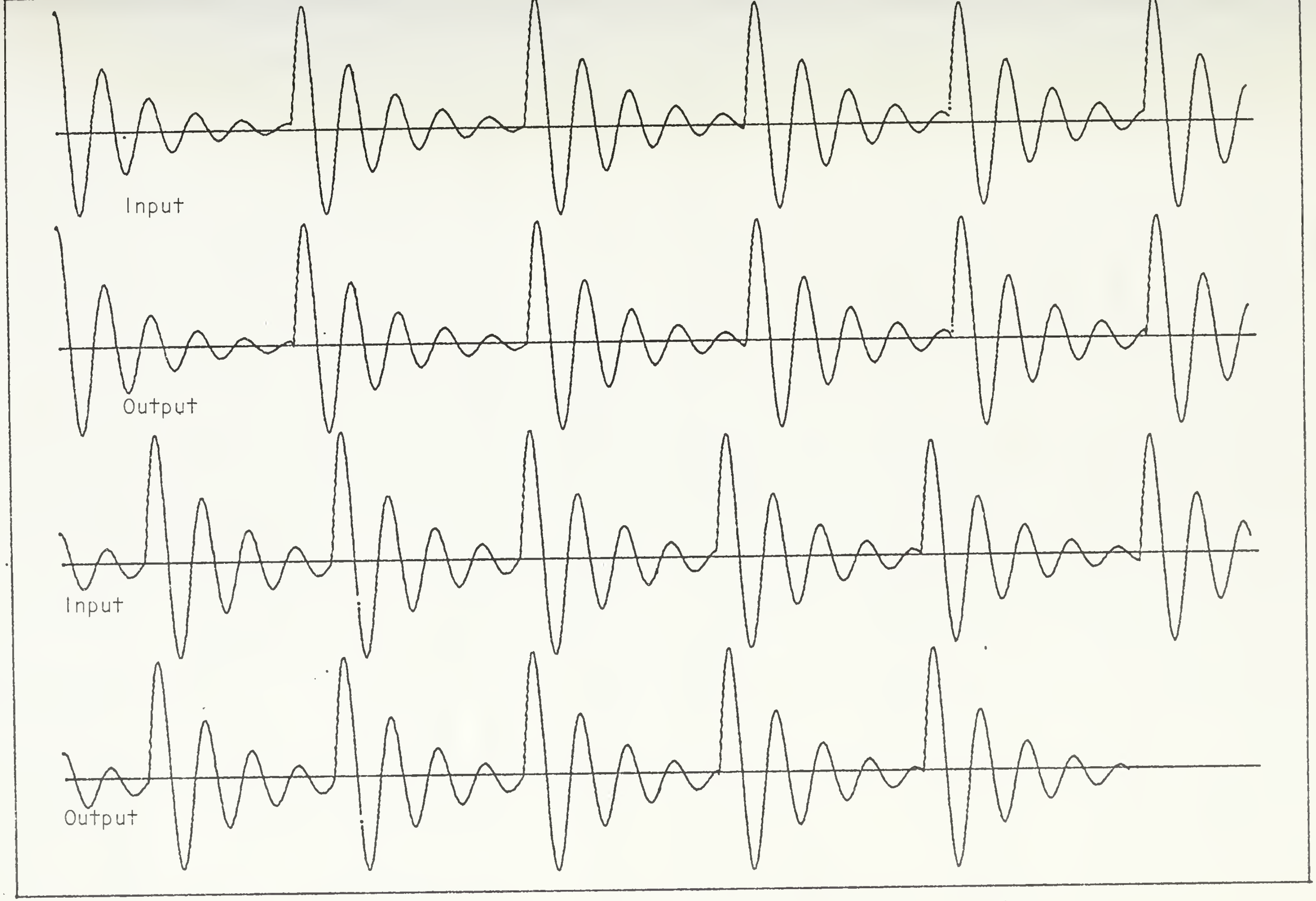


Figure 5-10 (con't.) Input and Output Waveforms - Adaptive System; Input Pass Test

A similar comparison can be made by viewing the spectra of the input and output waveforms. Figure 5-11 shows the spectrum of the input waveform, while figures 5-12, 5-13, and 5-14 show the output spectra of Shields' system, the adaptive overload system, and the adaptive system respectively. Figure 5-15 presents the error function for each of the systems.

For this test the system should be performing as an identity system. This says that the value for a in equation (5.11) should be equal to unity. Substituting this into equation (5.16):

$$E = -2 \log (1) = 0 \qquad (5.17)$$

This result is approached in all three systems, but the adaptive system comes the closest to matching the ideal error function for this test.

From all three presentations, the adaptive system clearly performs much better than the other two systems on the input pass test.

The second test performed was named the input reject test. The purpose of this test was to determine how well the systems would reject or attenuate an undesired speaker. This corresponds to the separational aspects of speech enhancement. The manner in which this test was conducted was simple. A waveform was generated with different pitch periods, and the pitch marks from that signal along with its pitch table were used to filter the original waveform.

The second waveform had a pitch period that also varied sinusoidally about some average value that was identical to the original

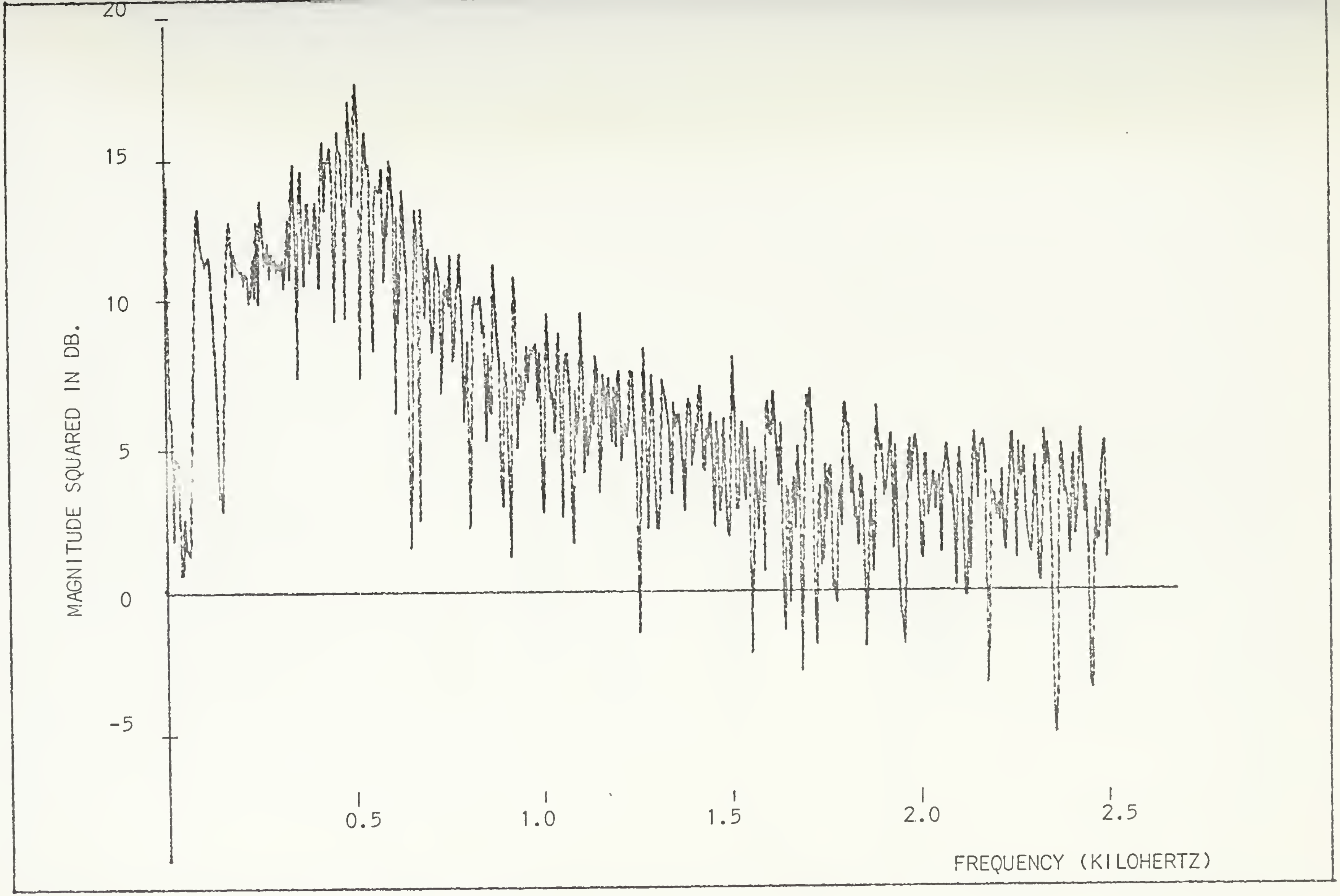


Figure 5-11 Spectrum of Input Waveform

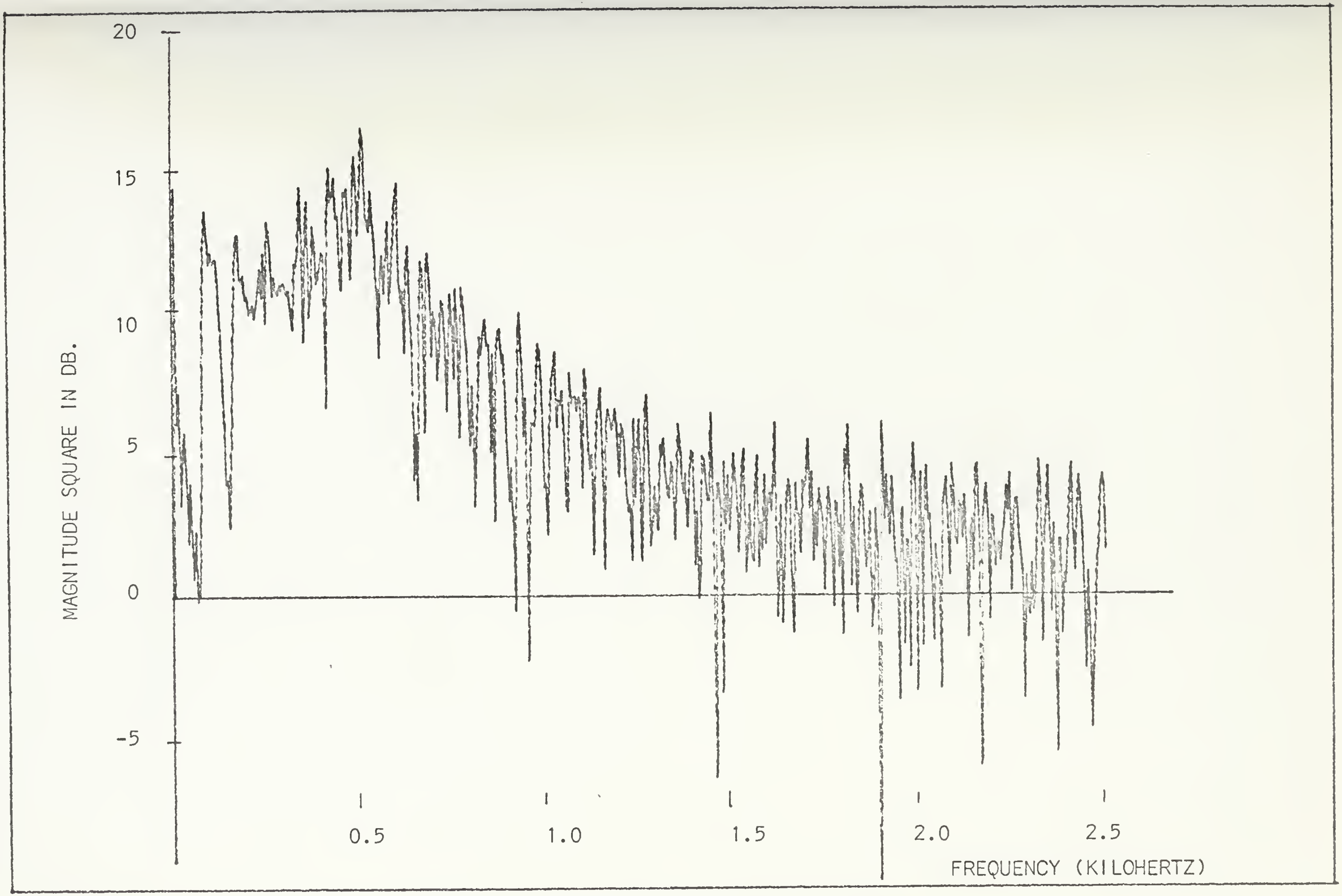


Figure 5-12 Output Spectrum - Shields' System; Input Pass Test

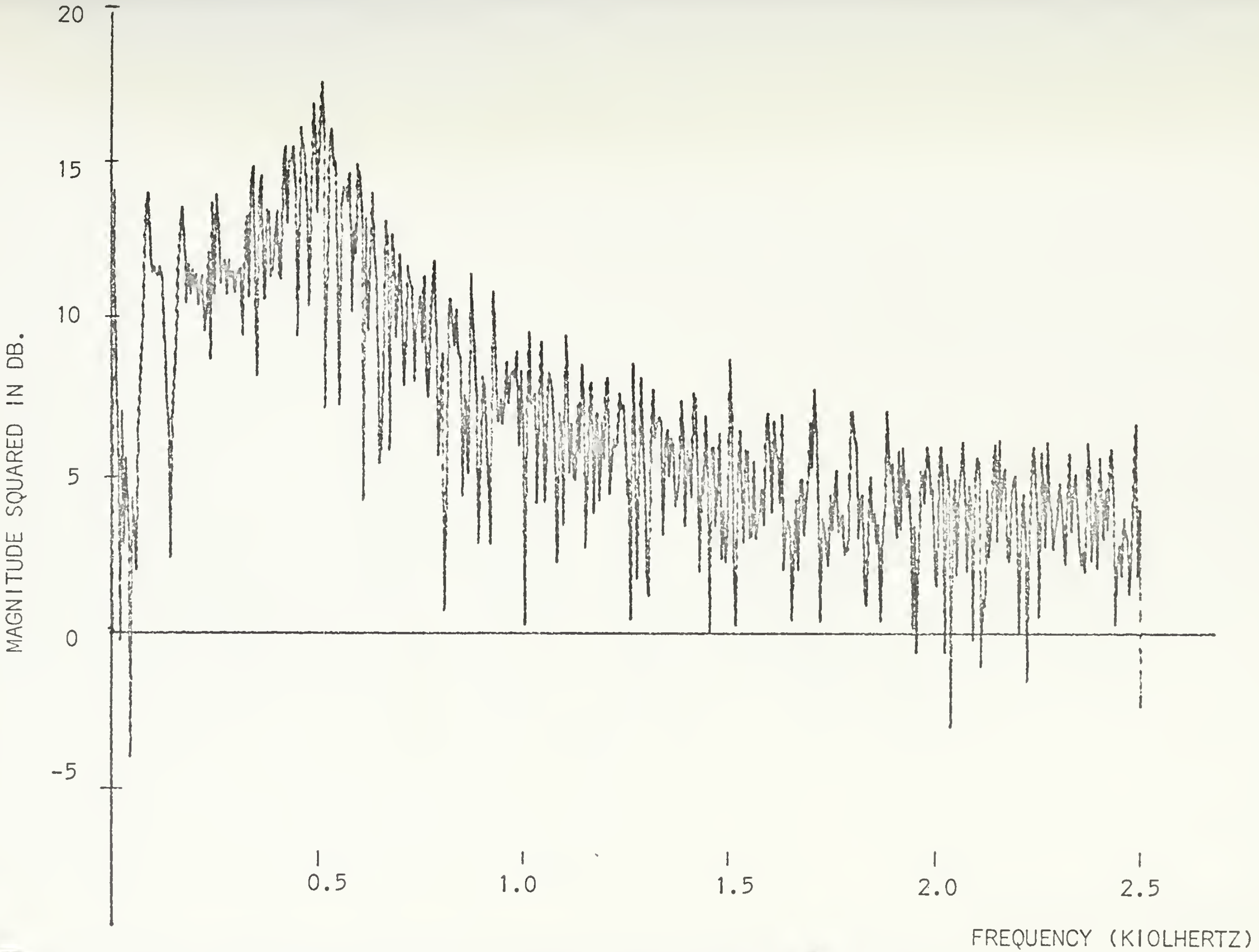


Figure 5-13 Output Spectrum - Adaptive Overload System; Input Pass Test

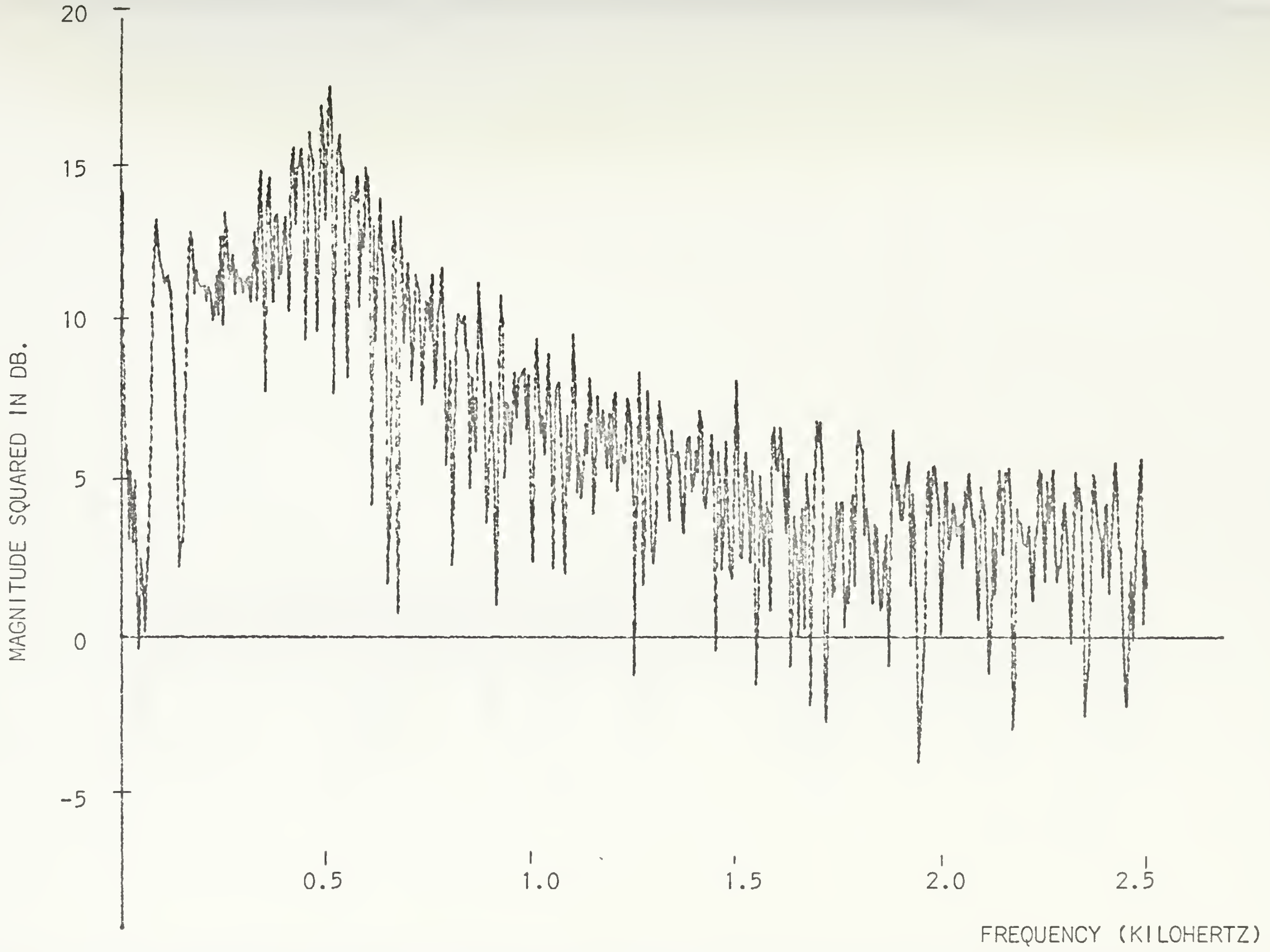


Figure 5-14 Output Spectrum - Adaptive System; Input Pass Test

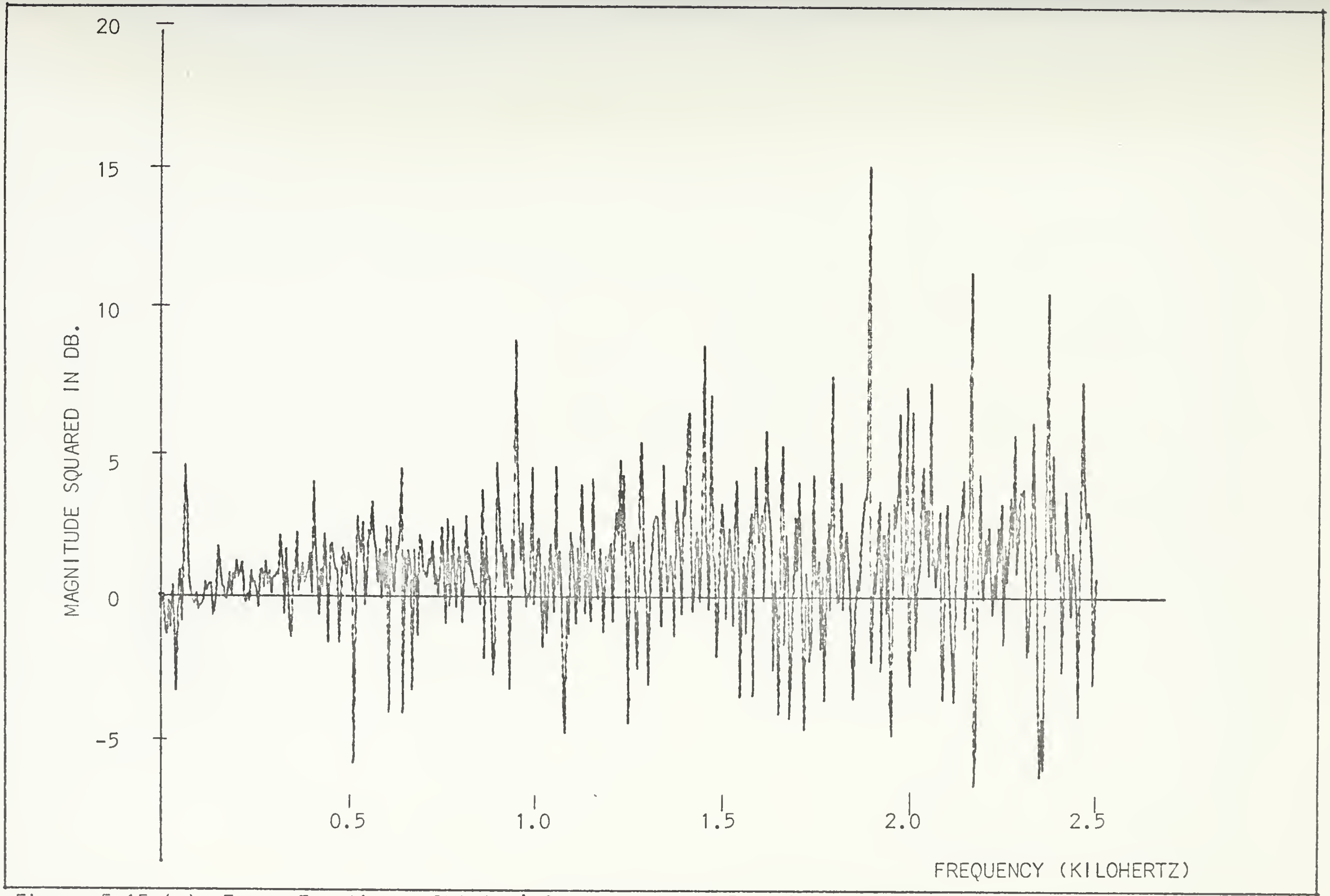


Figure 5-15 (a) Error Function - Shields' System; Input Pass Test

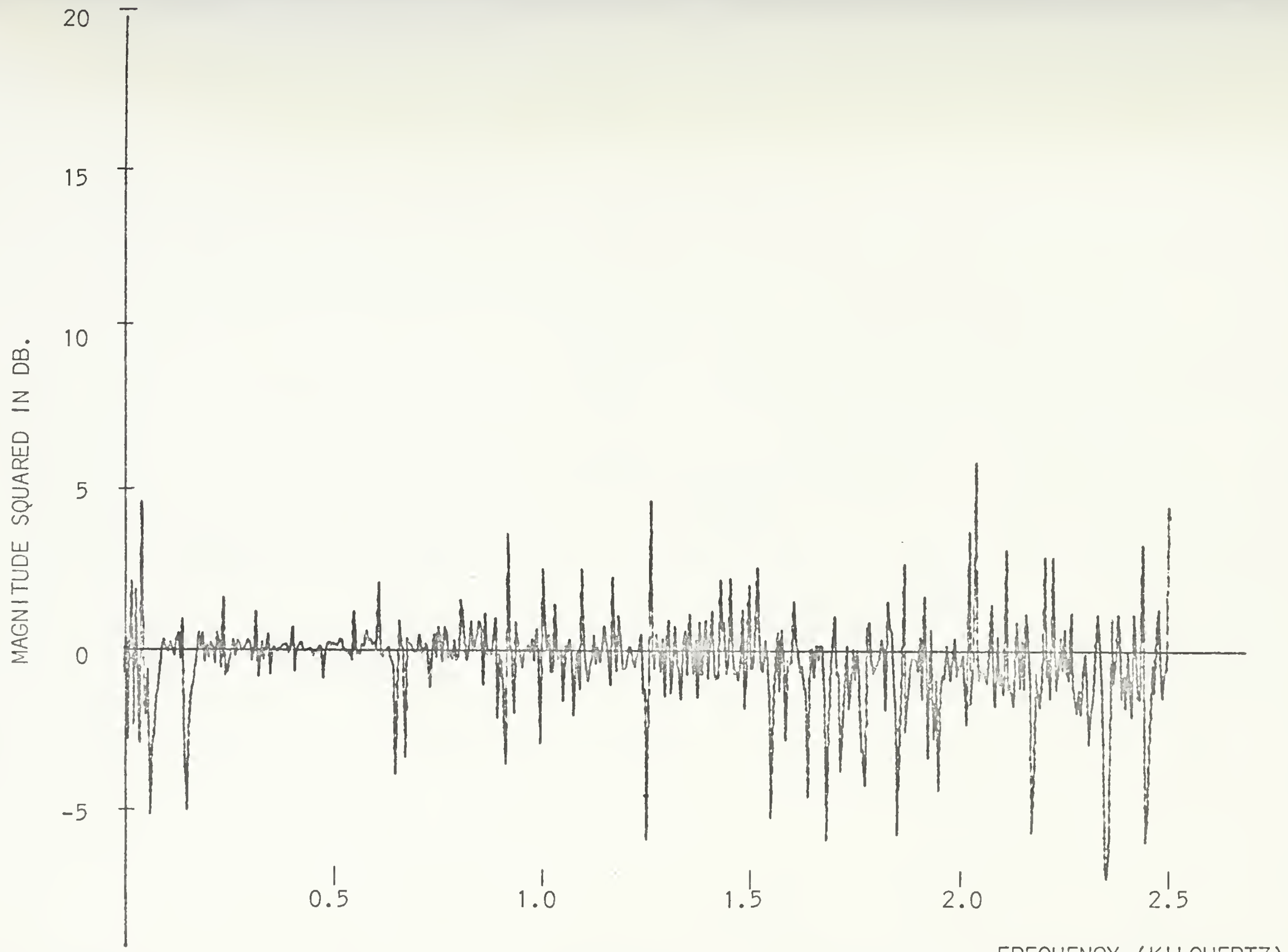


Figure 5-15 (b) Error Function - Adaptive Overload System; Input Pass Test

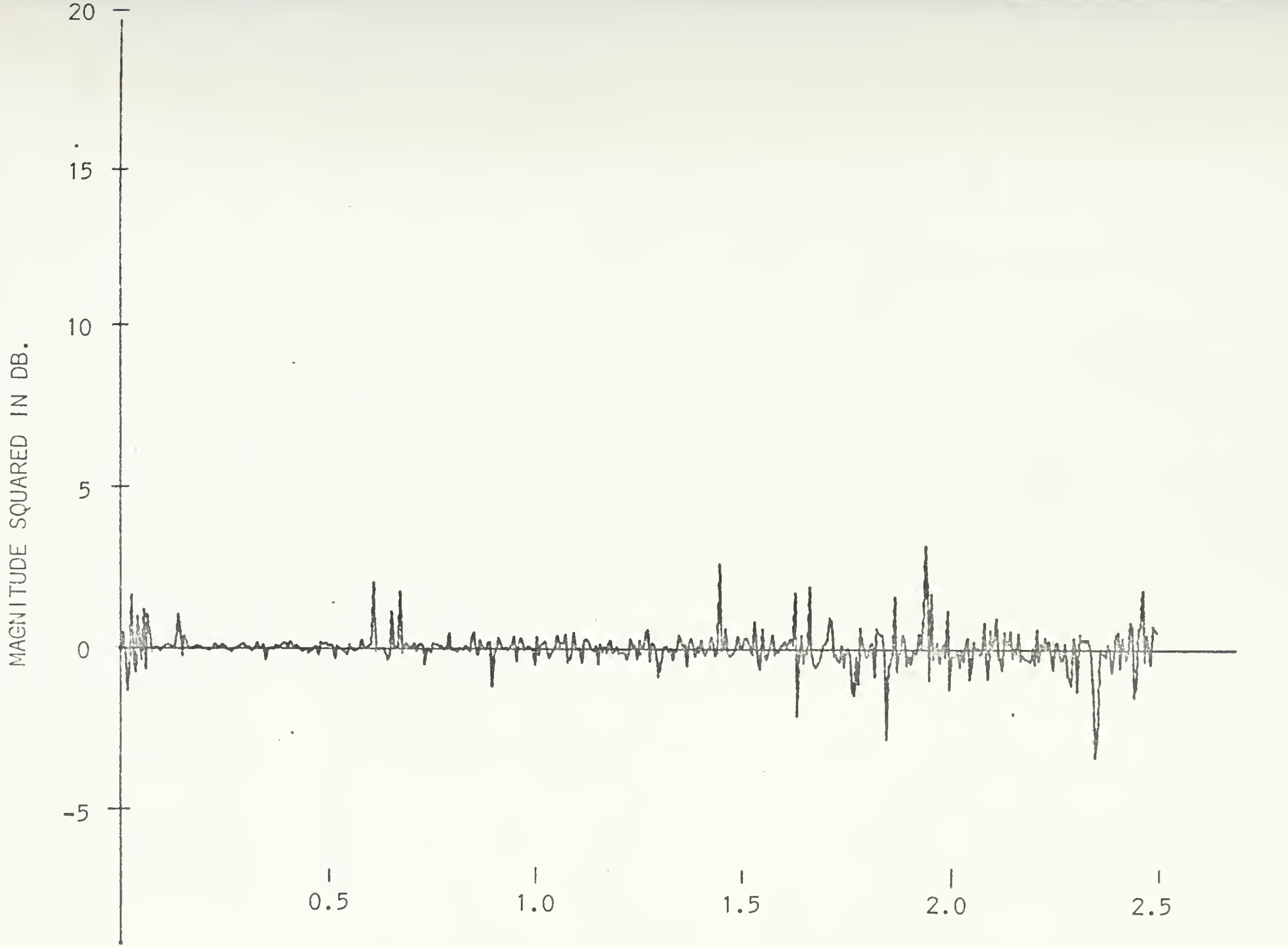


Figure 5-15 (c) Error Function - Adaptive System; Input Pass Test

waveform. The δ of the second waveform was changed in order to give the different set of pitch periods. It should be pointed out that this procedure allows for several of the corresponding periods to have the same pitch period. This is synonymous to a crossing of pitch contours in a pitch period versus time presentation. Ideally, the output from the systems should be zero, but this result could not be achieved.

Figures 5-16, 5-17, and 5-18 show the time domain results of the input reject test for Shields', the adaptive overload, and the adaptive systems respectively. From these presentations it is extremely difficult to establish which system is doing a better job of attenuating the input waveform. Each system has certain areas that it does well in, and areas that it does not do well in.

The frequency domain presentation does provide a better picture of how the systems performed on this test. The input signal spectrum has not been included in this series of figures because it would be the same spectrum as the one shown in figure 5-11. The output spectra for the Shields', the adaptive overload, and the adaptive system are presented in figures 5-19, 5-20, and 5-21 respectively. In all systems the general level of attenuation is on the order of five decibels.

The system shown in figure 5-7 may again be used to measure the performance. In equation (5.11), if the value of a is less than one, this corresponds to an attenuation of the input signal. Consider equation (5.16) again. If the value of a is less than unity, then, the logarithm of a will be negative making the value of E positive. As a approaches zero, the value of E approaches infinity. Therefore,

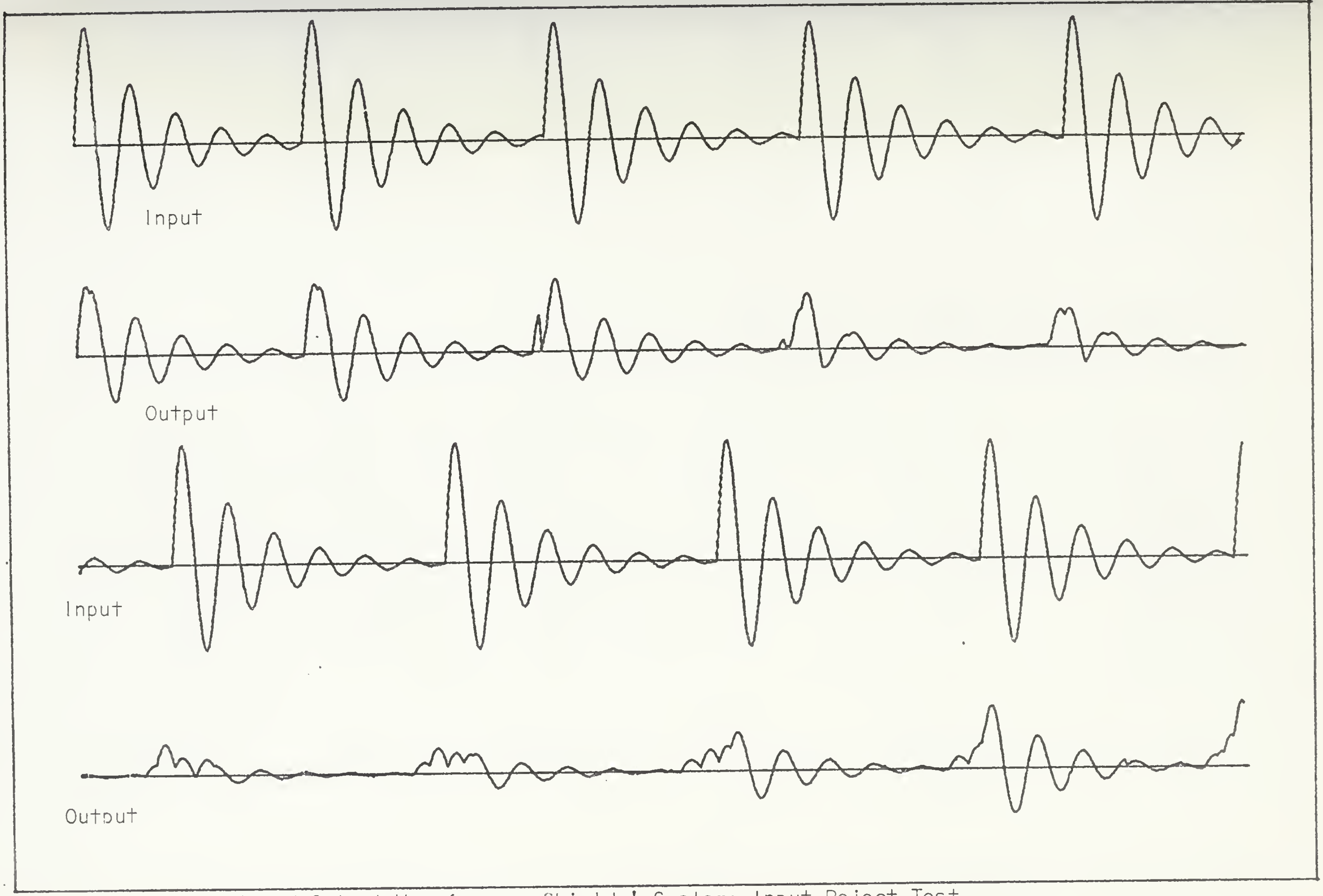


Figure 5-16 Input and Output Waveforms - Shields' System; Input Reject Test

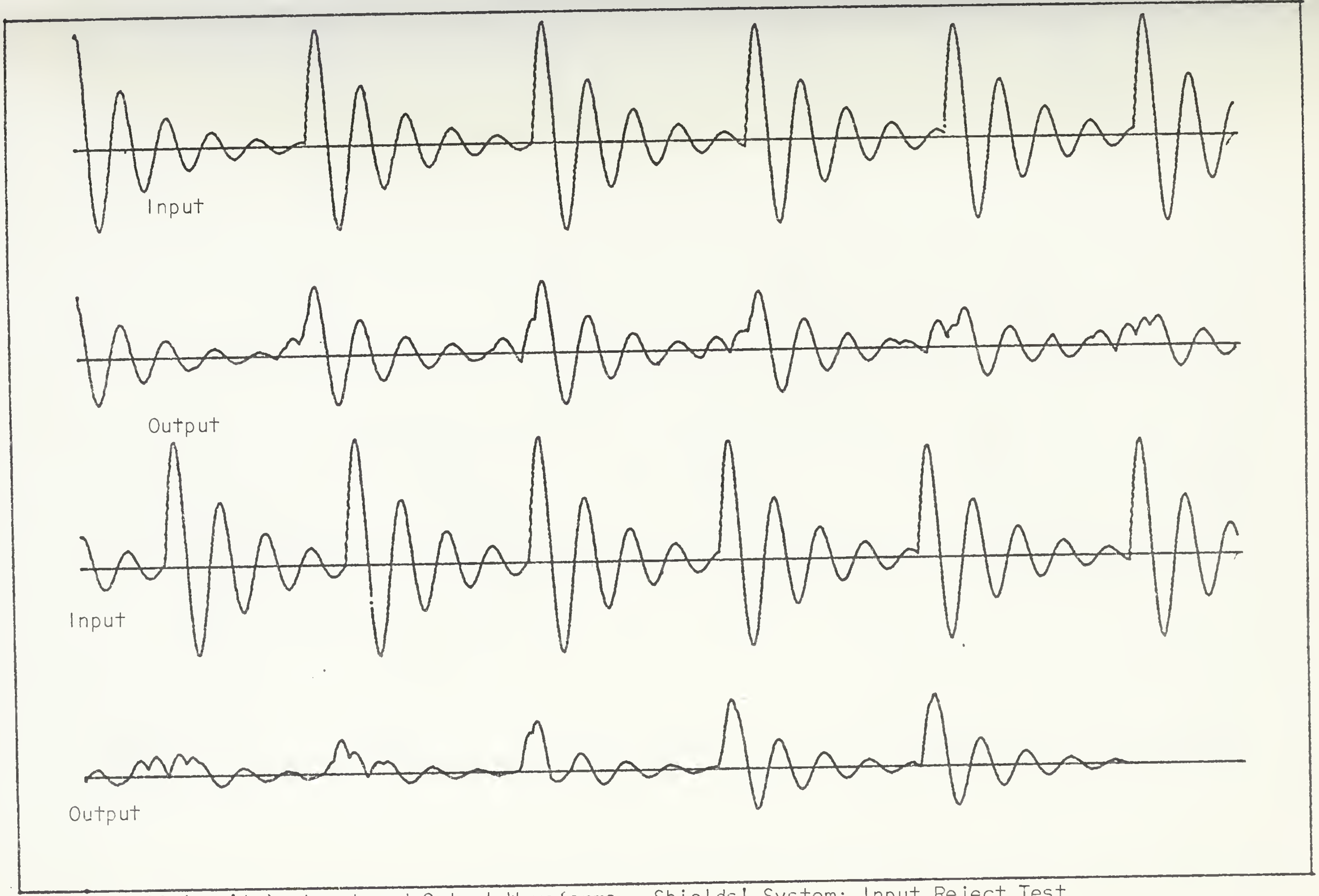


Figure 5-16 (con't.) Input and Output Waveforms - Shields' System; Input Reject Test

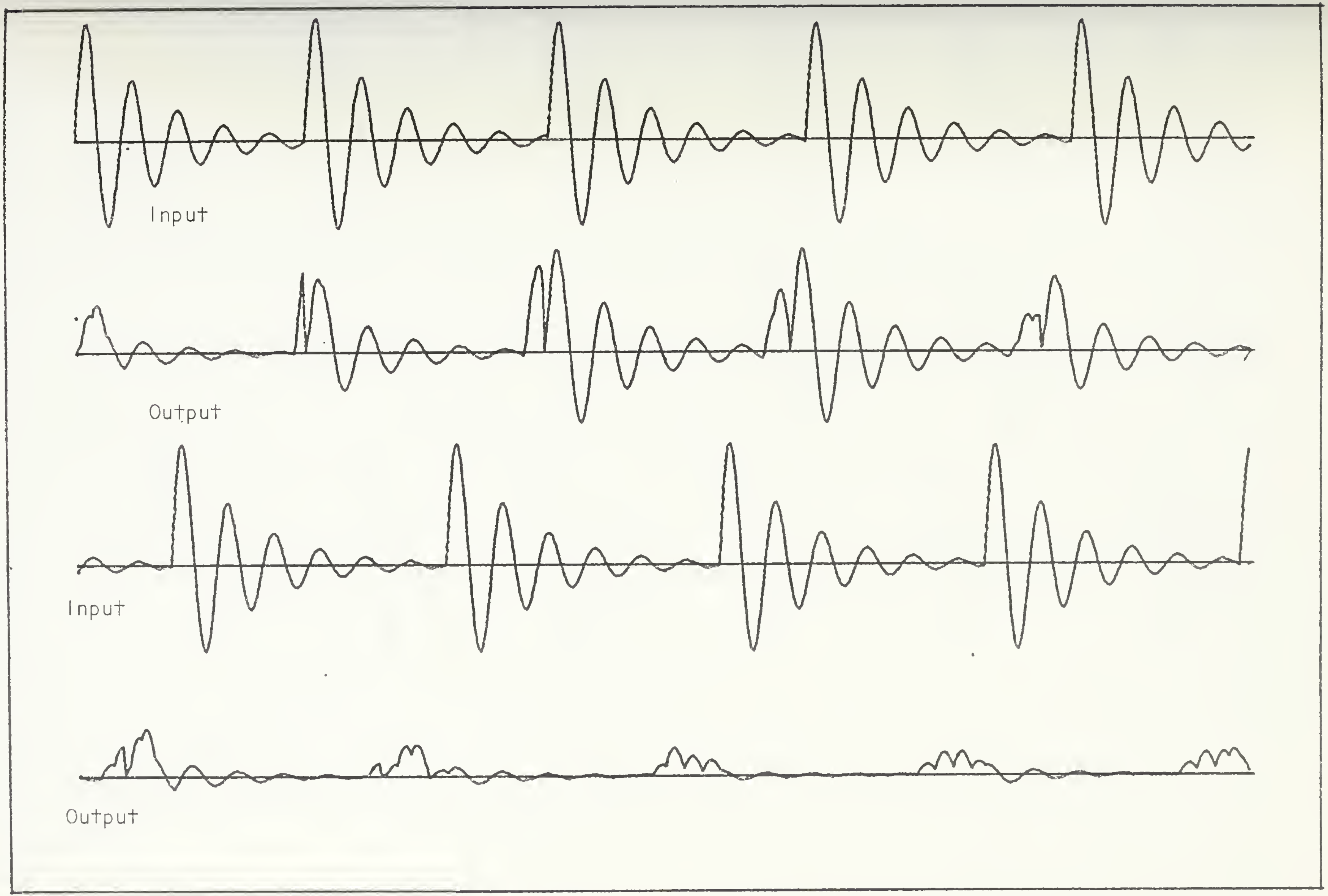


Figure 5-17 Input and Output Waveforms - Adaptive Overload System; Input Reject Test

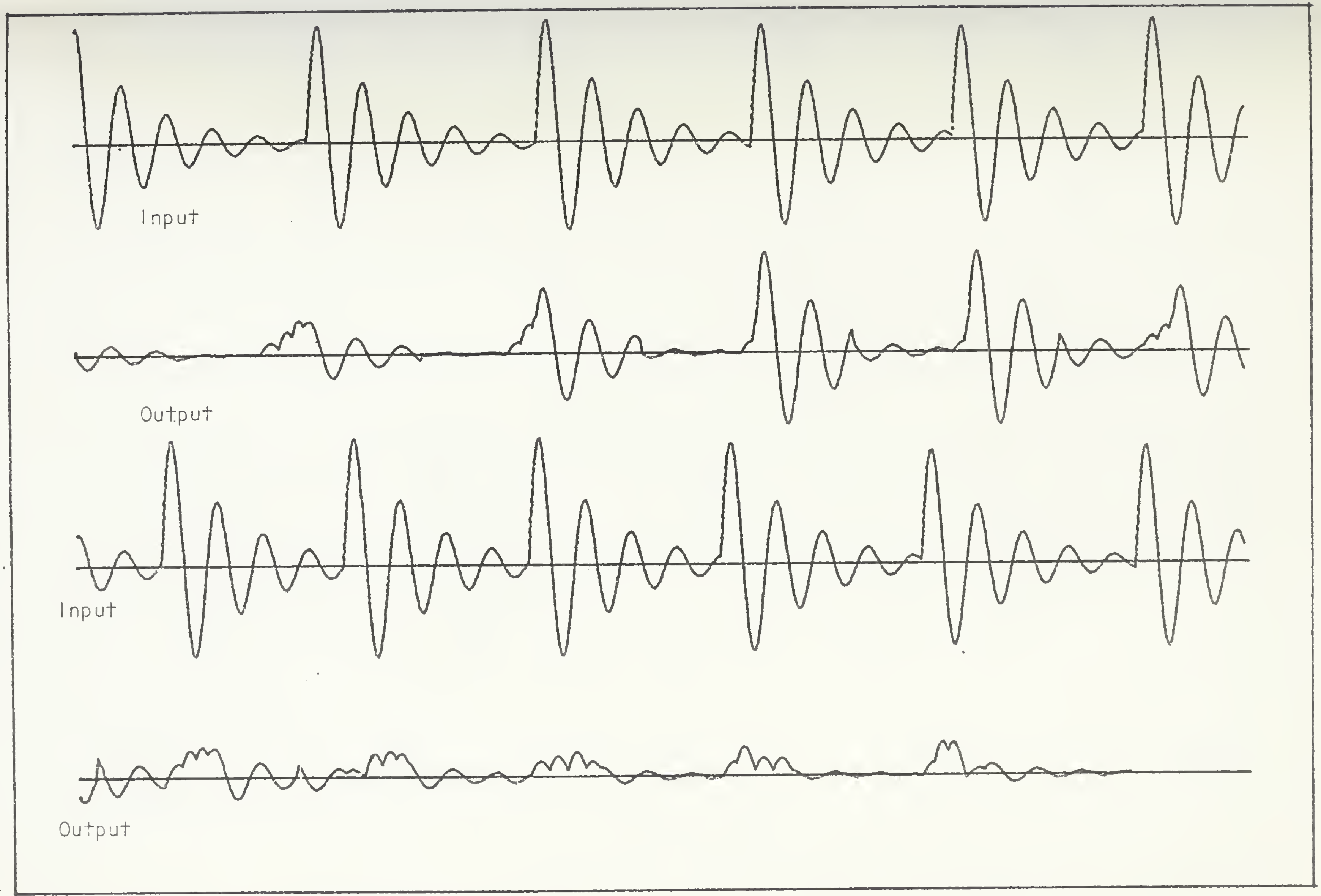


Figure 5-17 (con't.) Input and Output Waveforms - Adaptive Overload System; Input Reject System

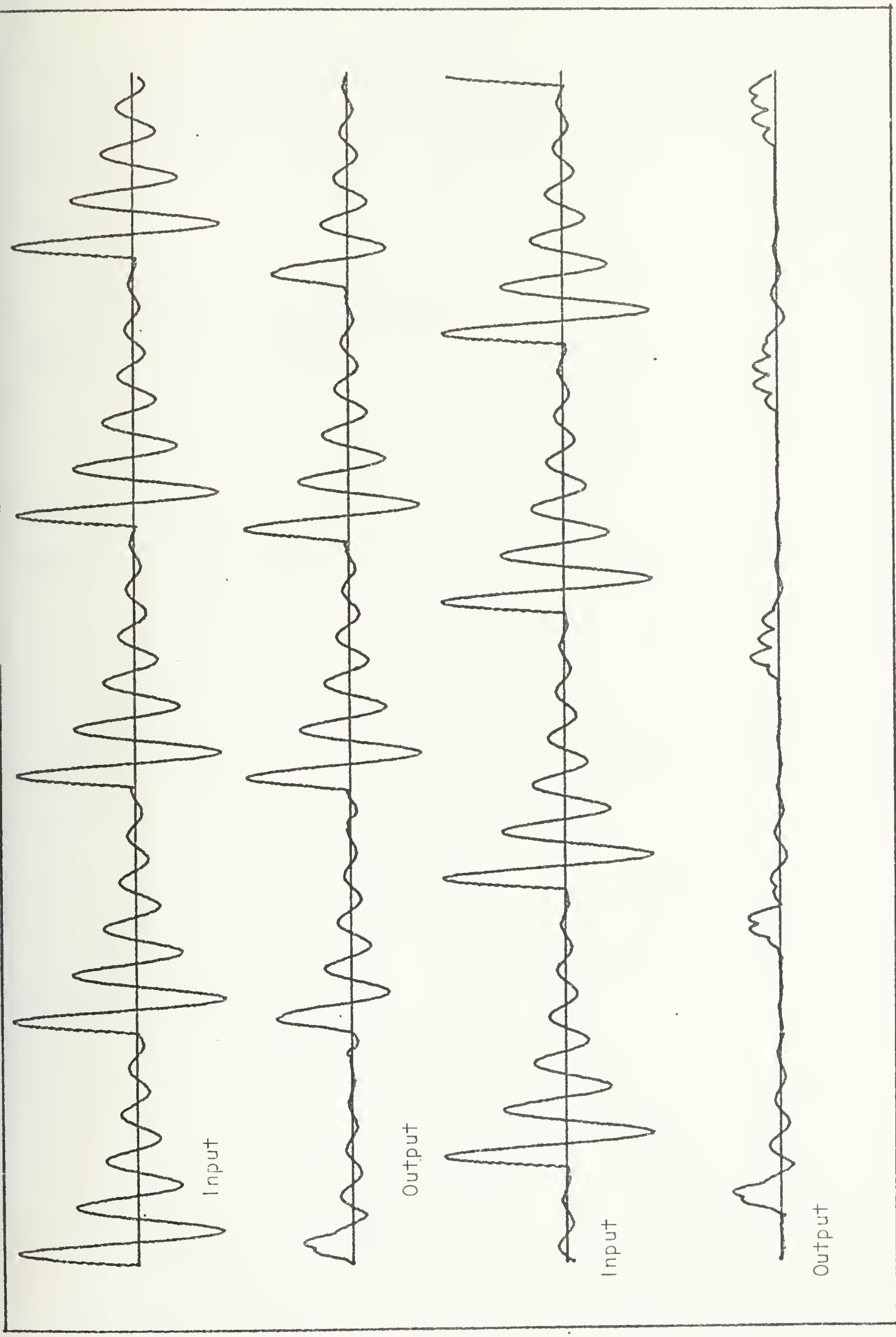


Figure 5-18 Input and Output Waveforms - Adaptive System; Input Reject Test

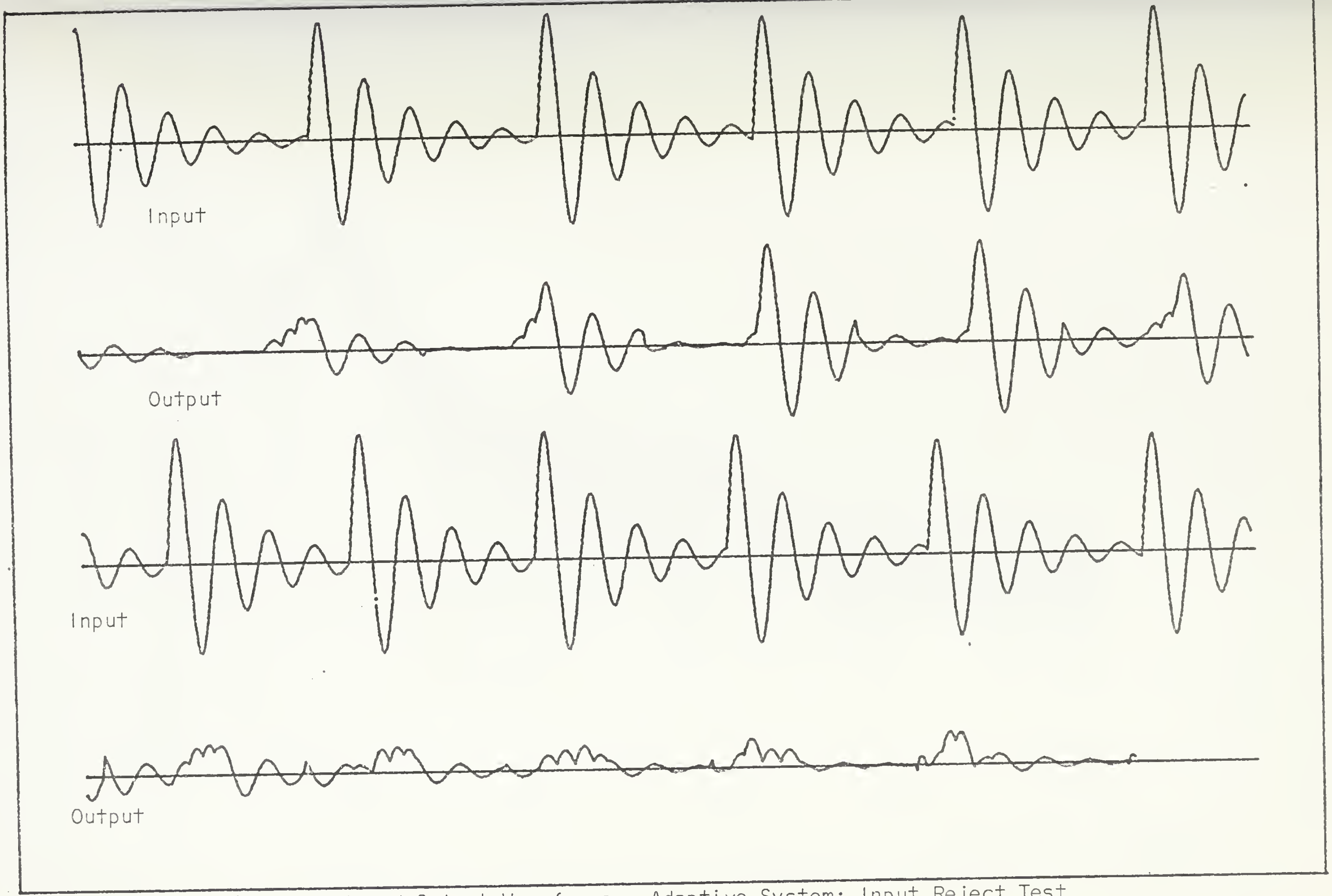


Figure 5-18 (con't.) Input and Output Waveforms - Adaptive System; Input Reject Test

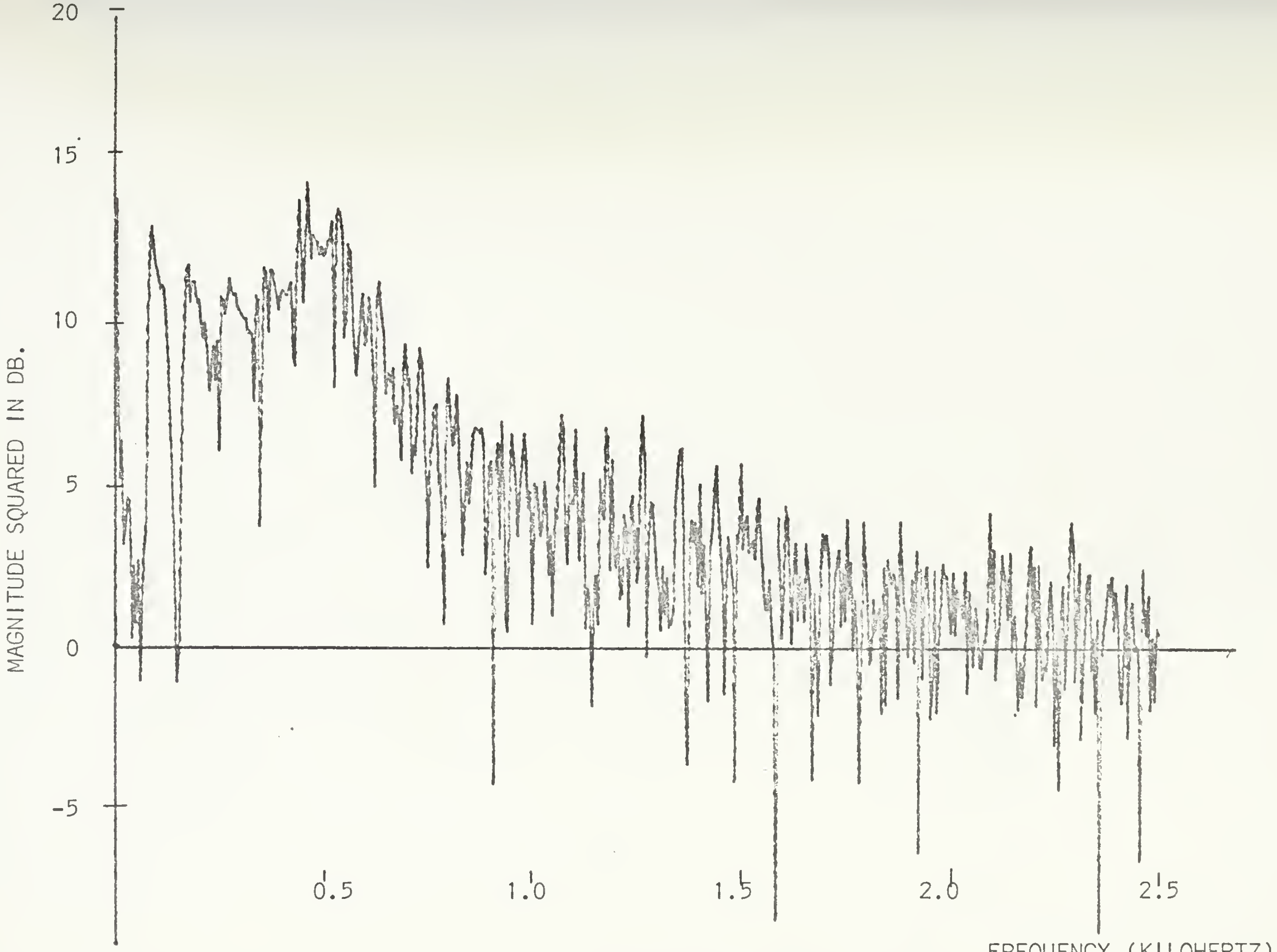


Figure 5-19 Output Spectrum - Shields' System; Input Reject Test

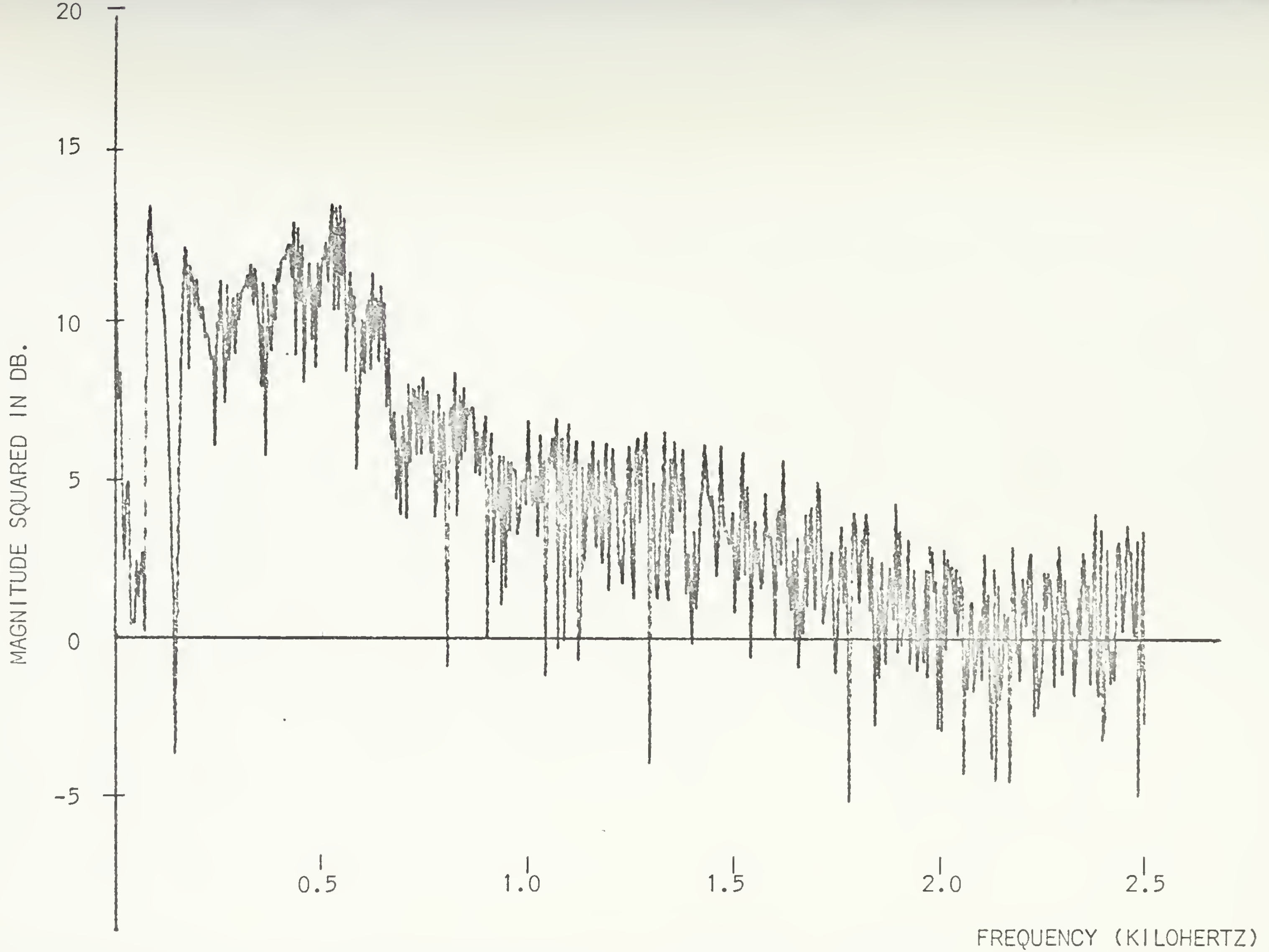


Figure 5-20 Output Spectrum - Adaptive Overload System; Input Reject Test

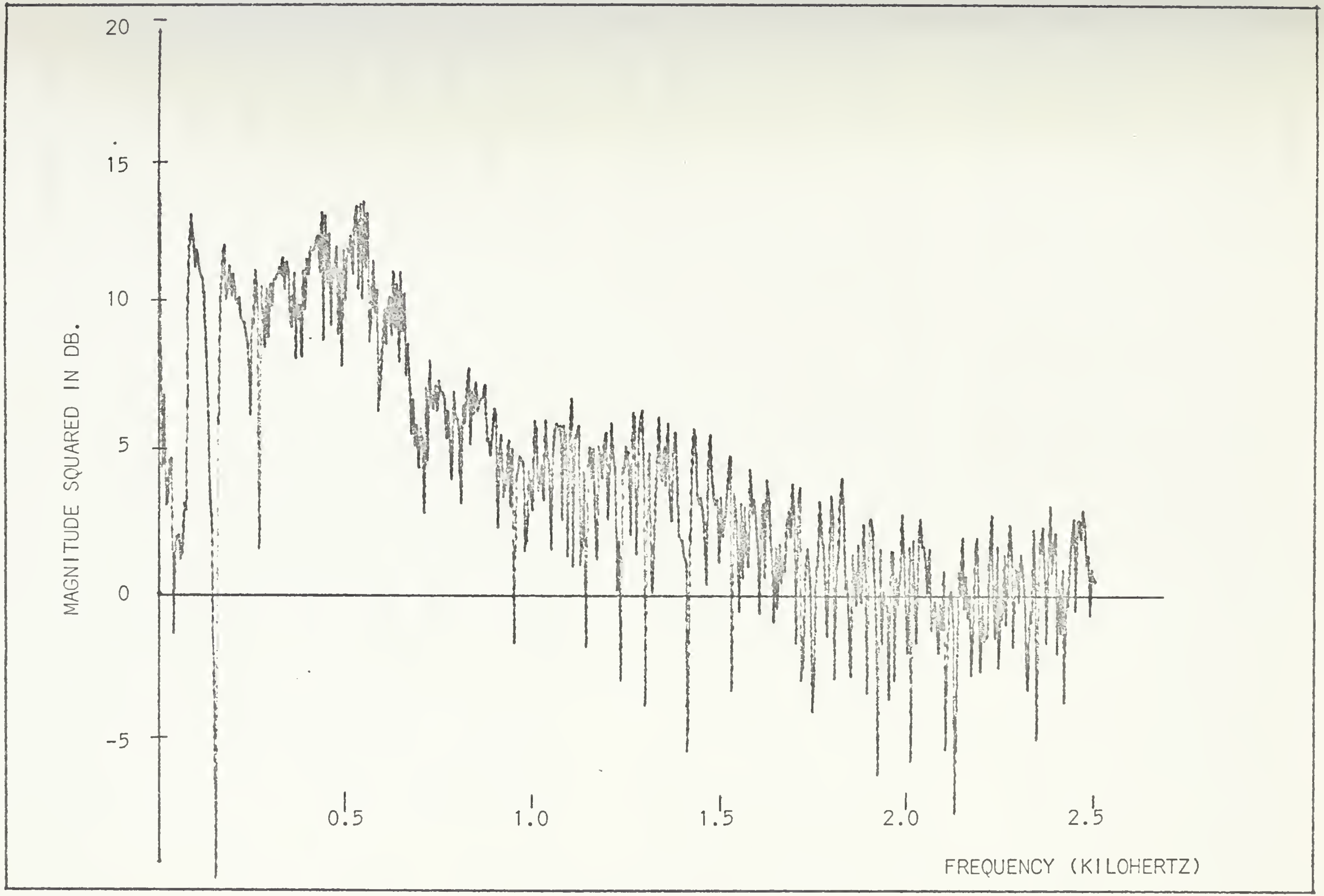


Figure 5-21 Output Spectrum - Adaptive System; Input Reject Test

in evaluating the system performance for this test, the magnitude of the error function is proportional to the amount of attenuation applied. The error functions are given in figure 5-22, and these error functions show several characteristics. The average level of the three error functions is approximately the same. Therefore, it may be stated that overall the adaptive systems seem to do no worse than the system proposed by Shields on the input reject test.

The next test concerned the performance of the systems in the presence of white noise. The white noise was generated by an analog noise source, lowpass filtered at 4.7 kilohertz, and then, sampled at 10 kilohertz. The gain of the noise was selected so that the signal-to-noise ratio would be much greater than one. The noise waveform was then added to the signal as described in figure 5-6 to produce the input waveform for the enhancement systems.

The conclusions from Shields thesis were that the system did not perform well in the presence of white noise. The observation that the comb filter changed the wideband noise into noise that was highly harmonic is exemplified by figure 5-23. The output waveform in figure 5-23 is quite different from the input signal. The same statement may be made concerning the output of the adaptive overload system, which is shown in figure 5-24. The output of the adaptive system as shown in figure 5-25 is quite different from the outputs of the other two systems. The input and output waveforms are almost identical, and this is quite encouraging. The adaptive system passes the signal and noise combination almost exactly. In the adaptive overload and in

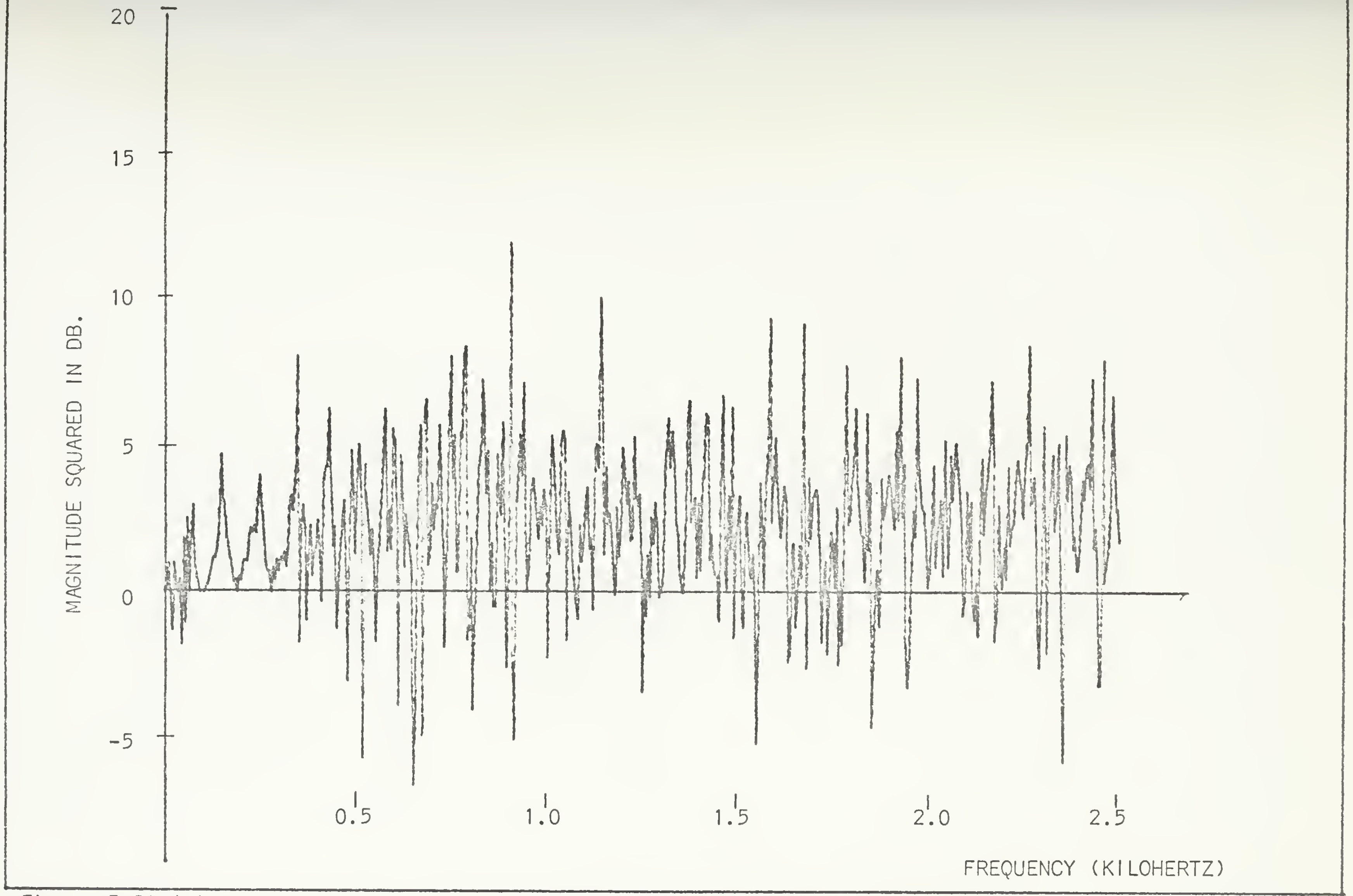


Figure 5-22 (a) Error Function - Shields' System; Input Reject Test

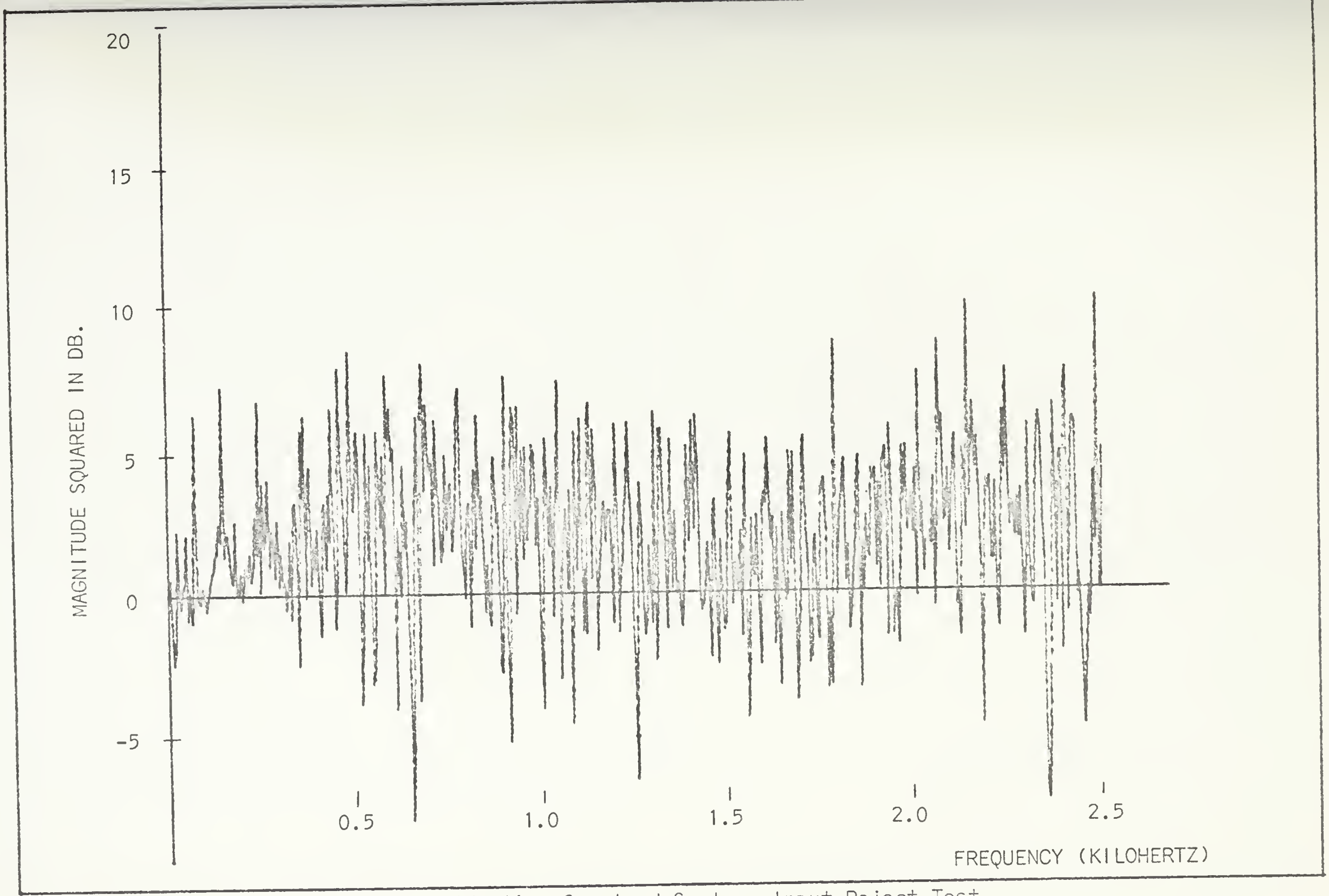


Figure 5-22 (b) Error Function - Adaptive Overload System; Input Reject Test

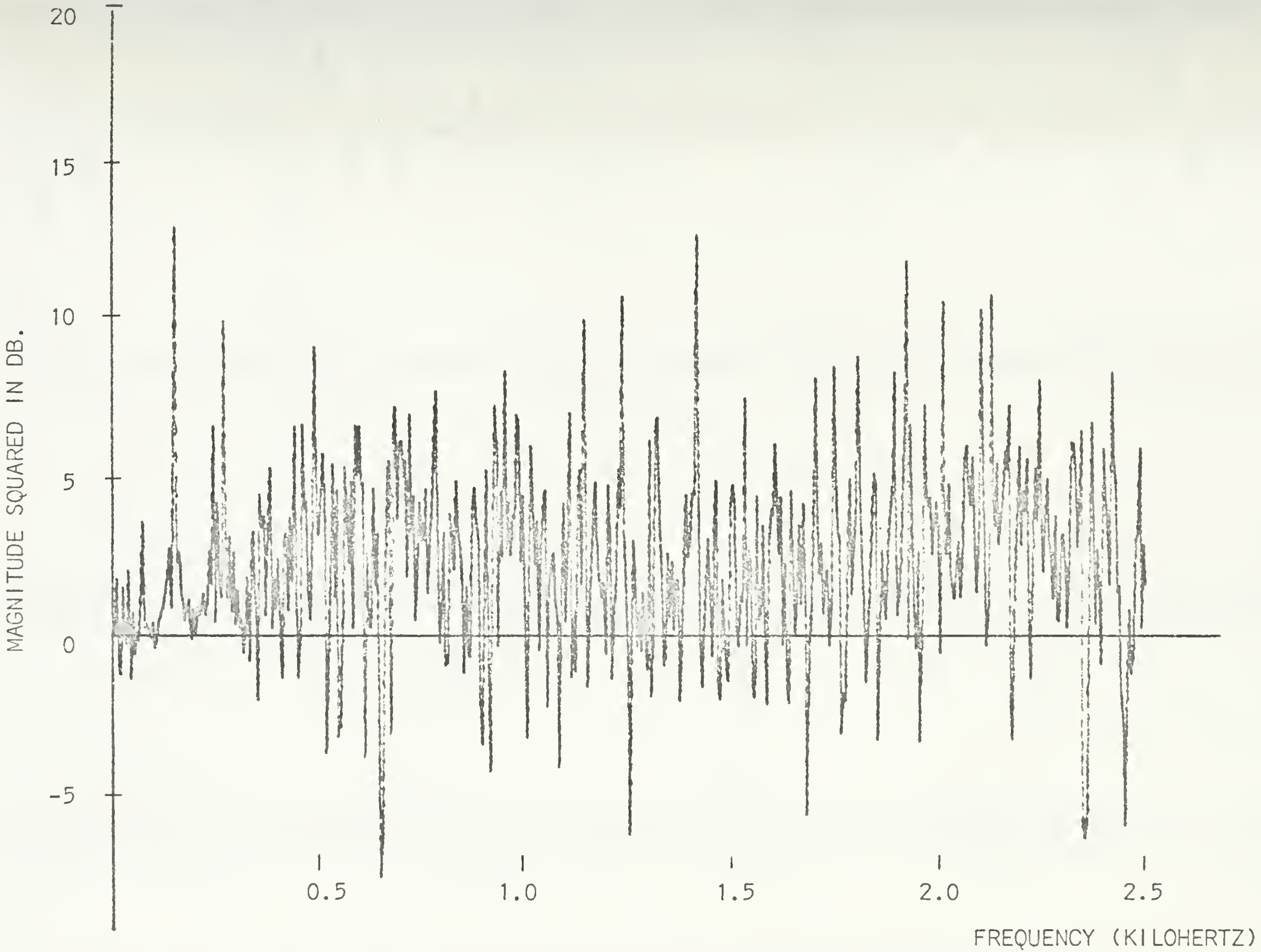


Figure 5-22 (c) Error Function - Adaptive System; Input Reject Test

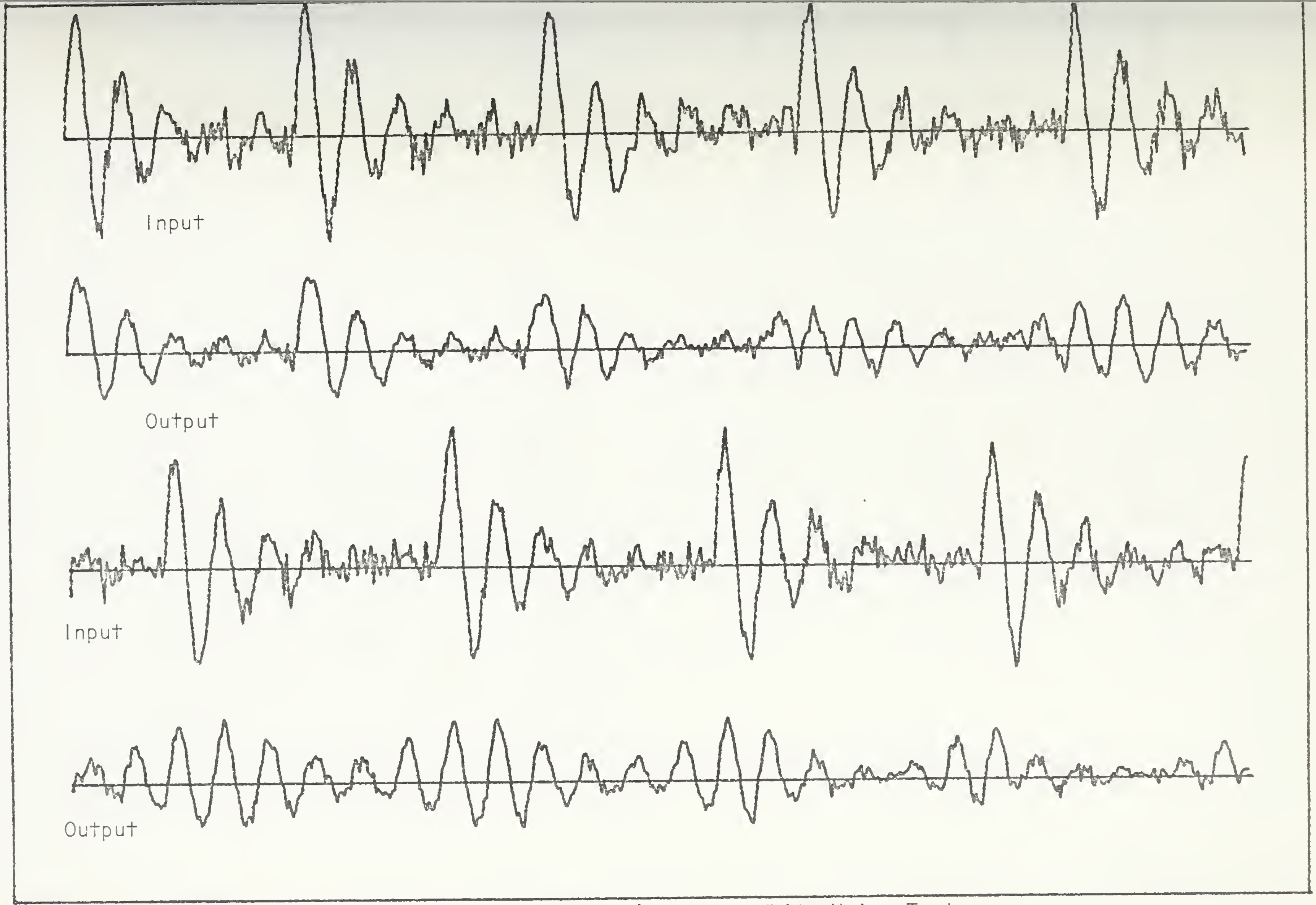


Figure 5-23 Input and Output Waveforms - Shields' System; White Noise Test

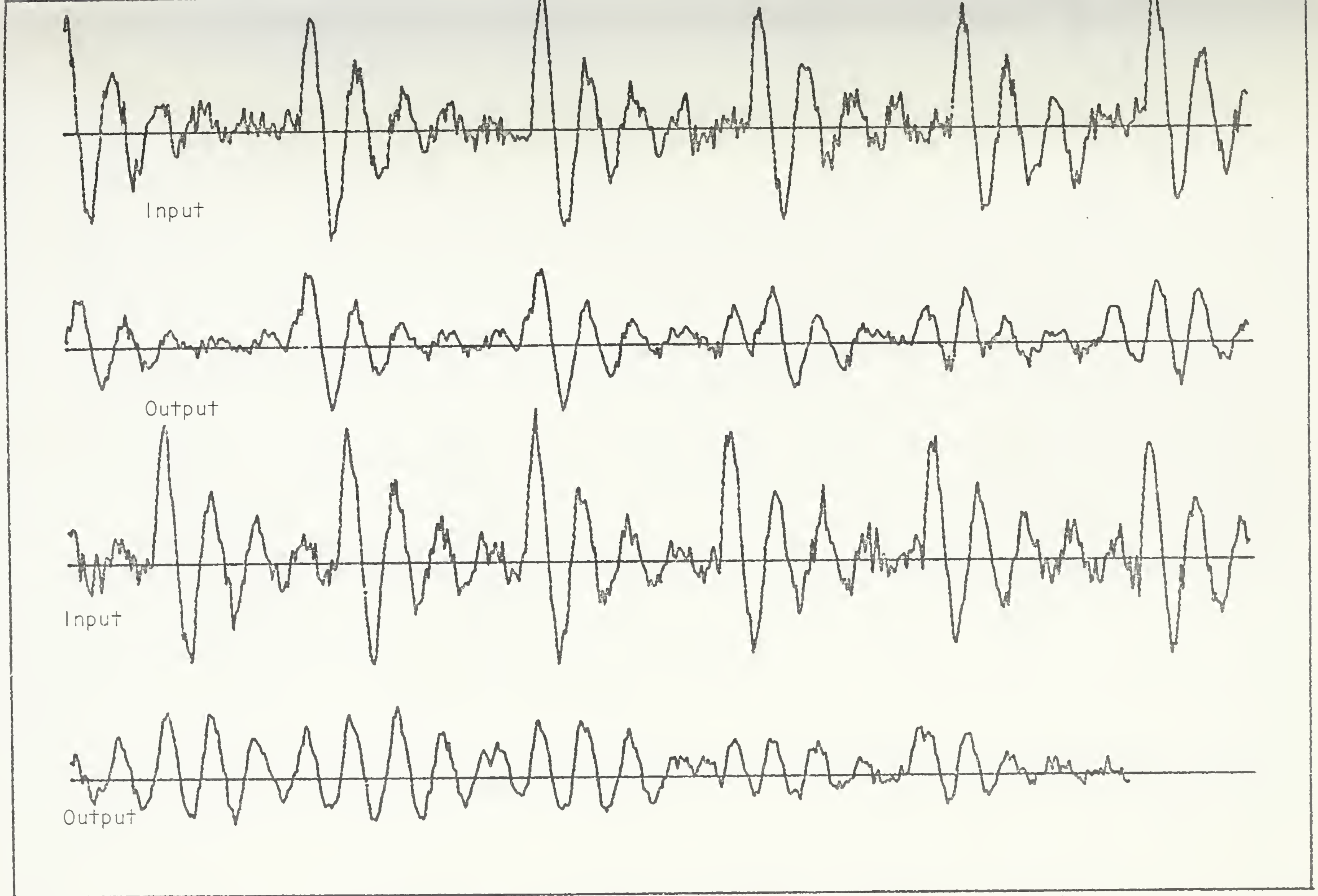


Figure 5-23 (con't.) Input and Output Waveforms - Shields' System; White Noise Test

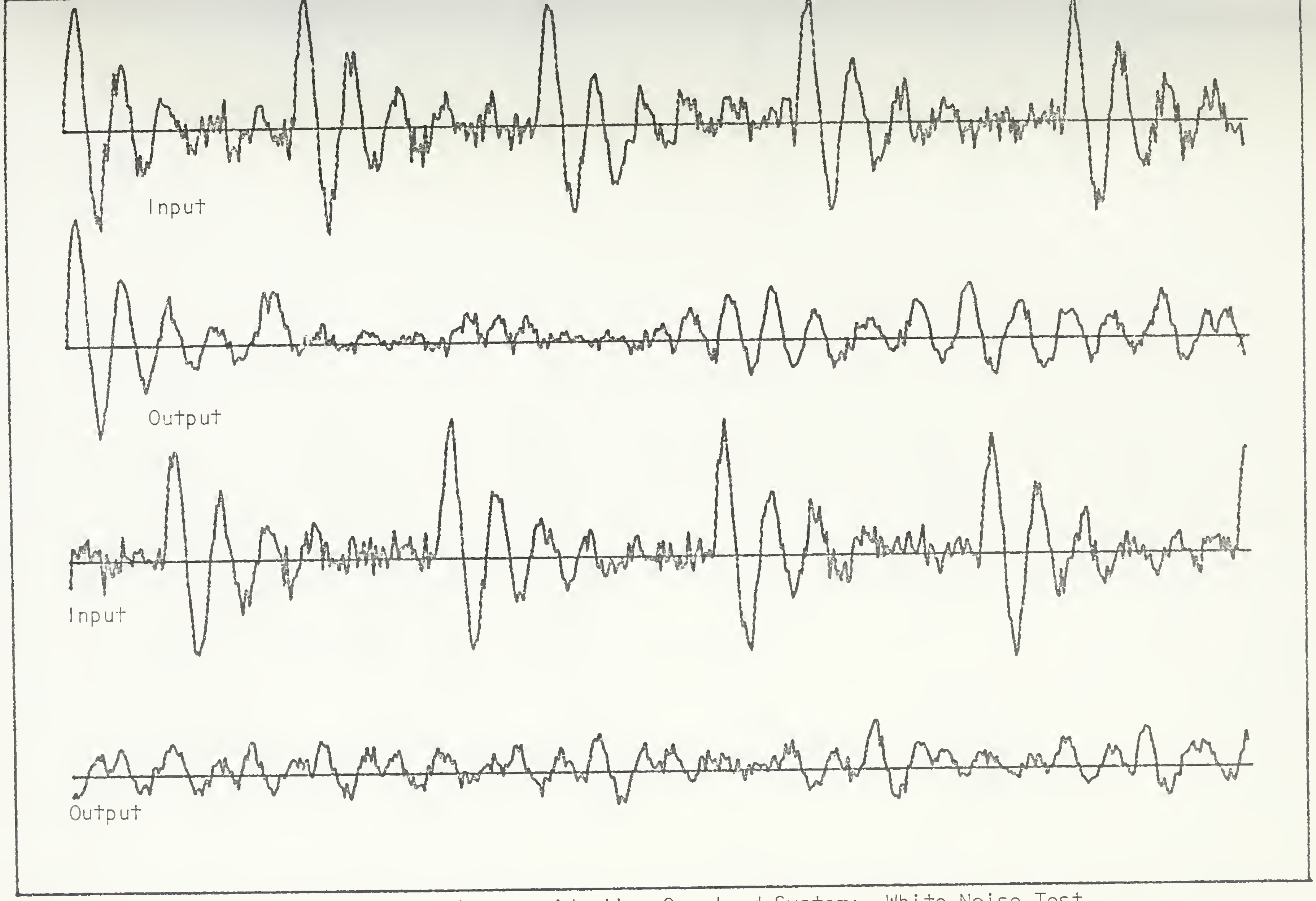


Figure 5-24 Input and Output Waveforms - Adaptive Overload System; White Noise Test

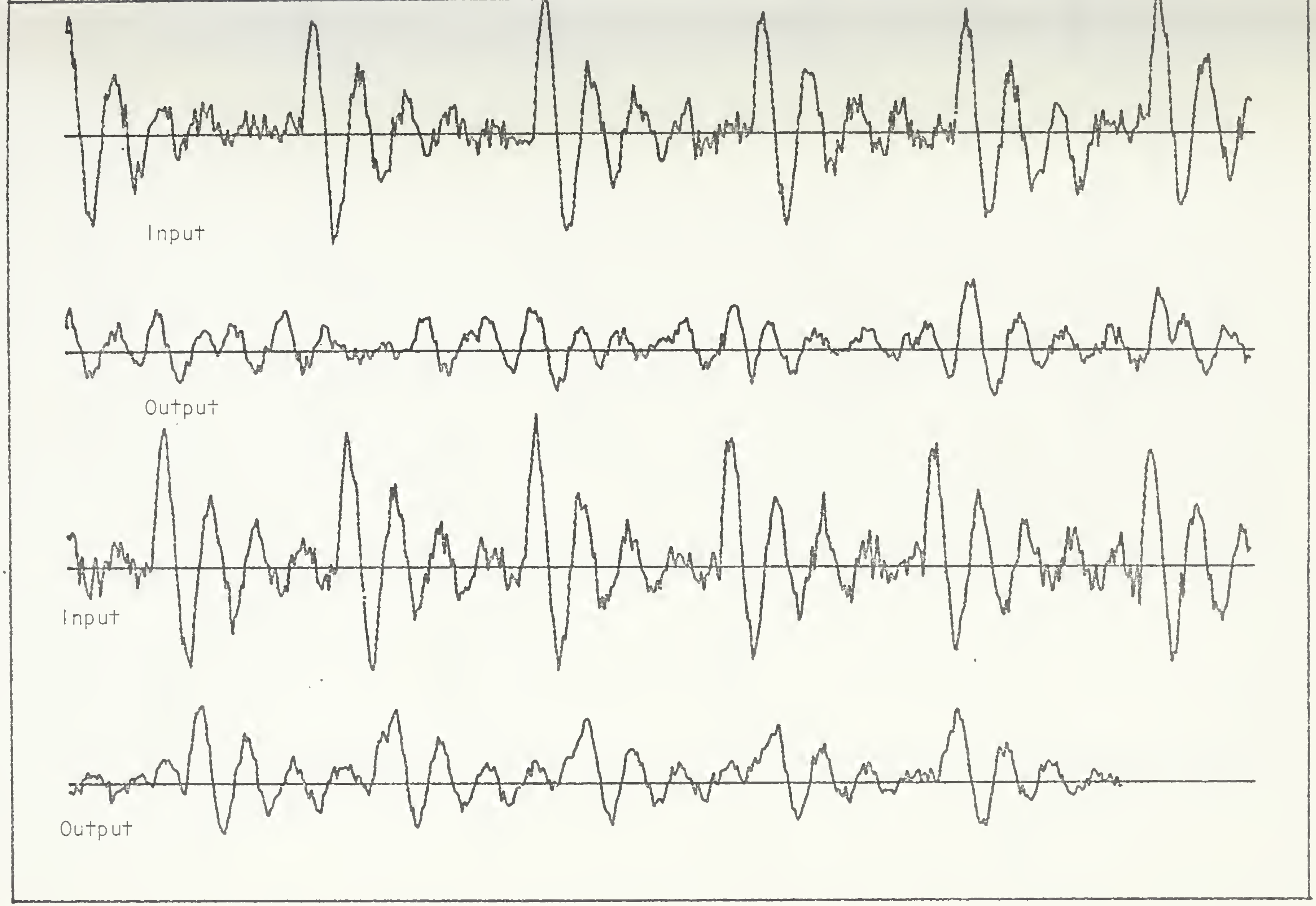


Figure 5-24 (con't.) Input and Output Waveforms - Adaptive Overload System; White Noise Test

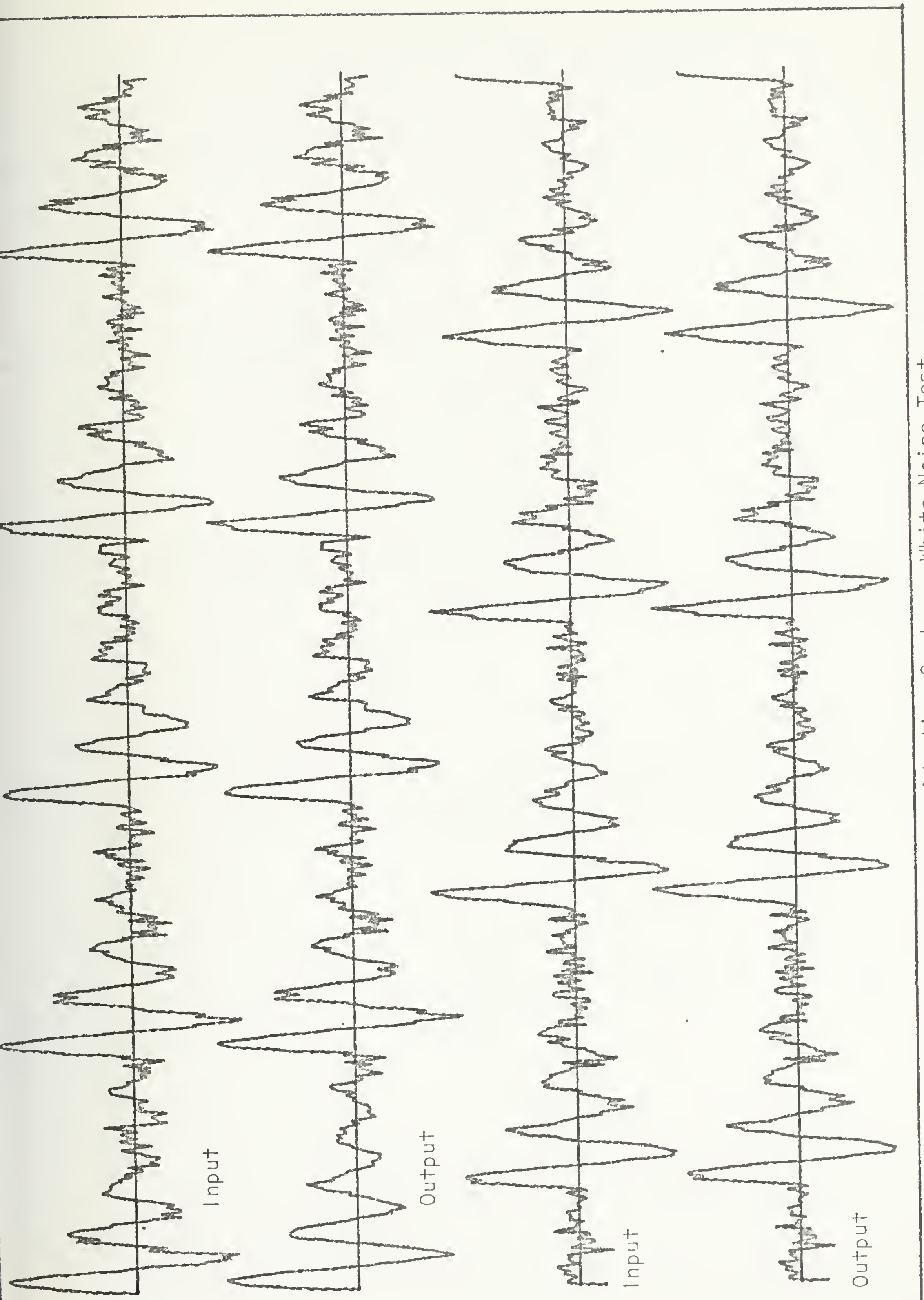


Figure 5-25 Input and Output Waveforms - Adaptive System; White Noise Test

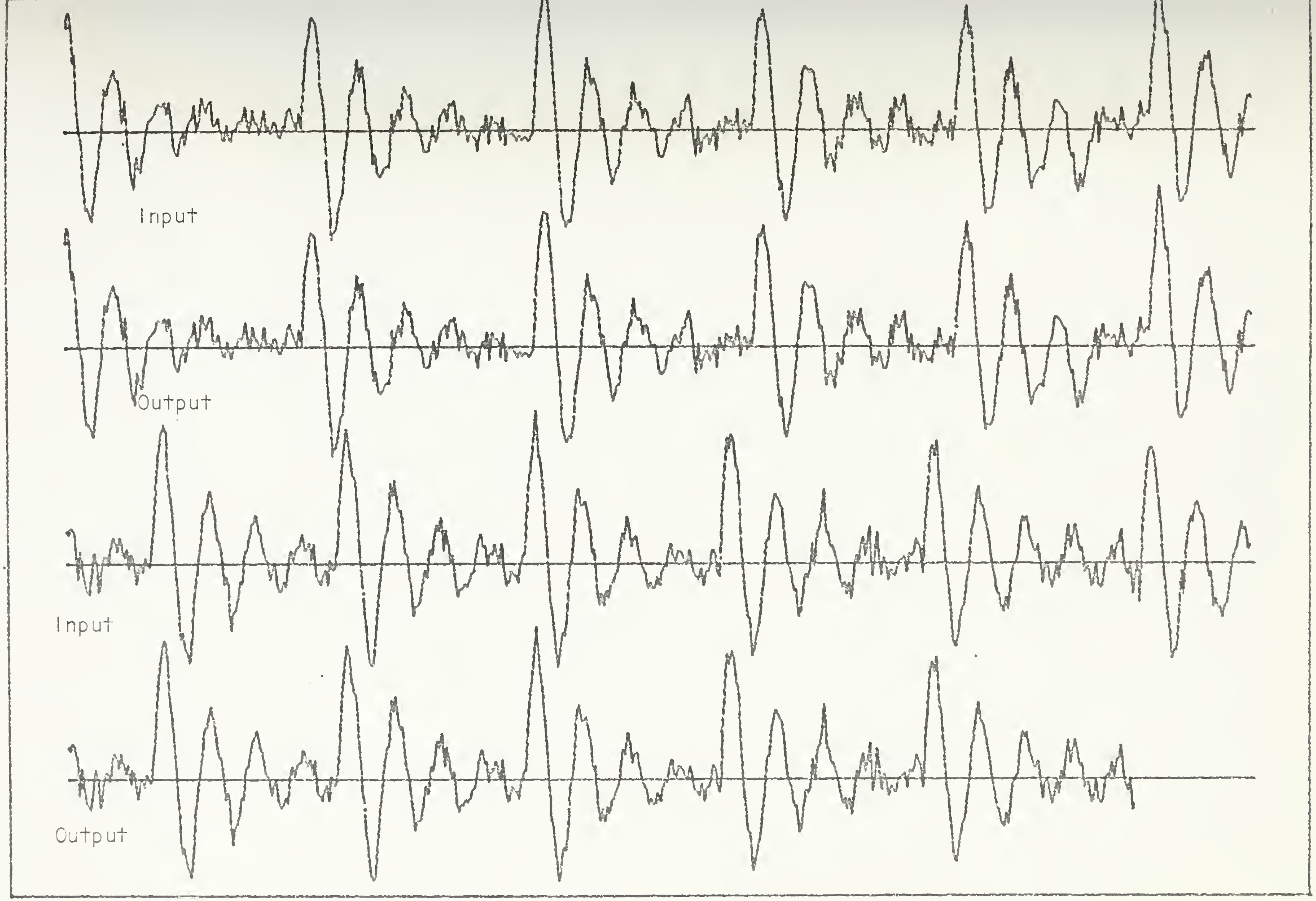


Figure 5-25 (con't.) Input and Output Waveforms - Adaptive System; White Noise Test

Shields' system, the output waveform is distorted immensely due to the white noise. On the other hand, the adaptive system's output is noisy, but the distortion of the other two systems is not present. Figure 5-26 shows the noiseless input signal and the output of the adaptive system with seven coefficients in the filter. It can be seen that the effects of increasing the number of coefficients in this case is negligible.

The spectrum of the noise used for this test is shown in figure 5-27, and the spectrum of the input signal added to the noise is shown in figure 5-28. Figure 5-29 is the spectrum of the output from Shields' system for the case of the signal with additive white noise. It can be seen that the spectrum is composed of distinct harmonic bands that were not as prevalent in the other cases. The areas outside the harmonic bands have been lost in the filtering process, therefore, causing distortion in the output. The output spectrum for the adaptive overload system, figure 5-30, resembles the input more closely than Shields' system, but there is some energy getting lost as the level of the spectrum is less than that of the input's spectrum. The adaptive system's output spectrum, figure 5-31, has only minor differences from the spectrum of the input. Figure 5-32 shows the error functions of the outputs in the same manner that they have been presented in the past.

There is little doubt that the adaptive system performs better than the other two systems from the standpoint of this particular test. The adaptive system provides an output signal that may be noisy, but

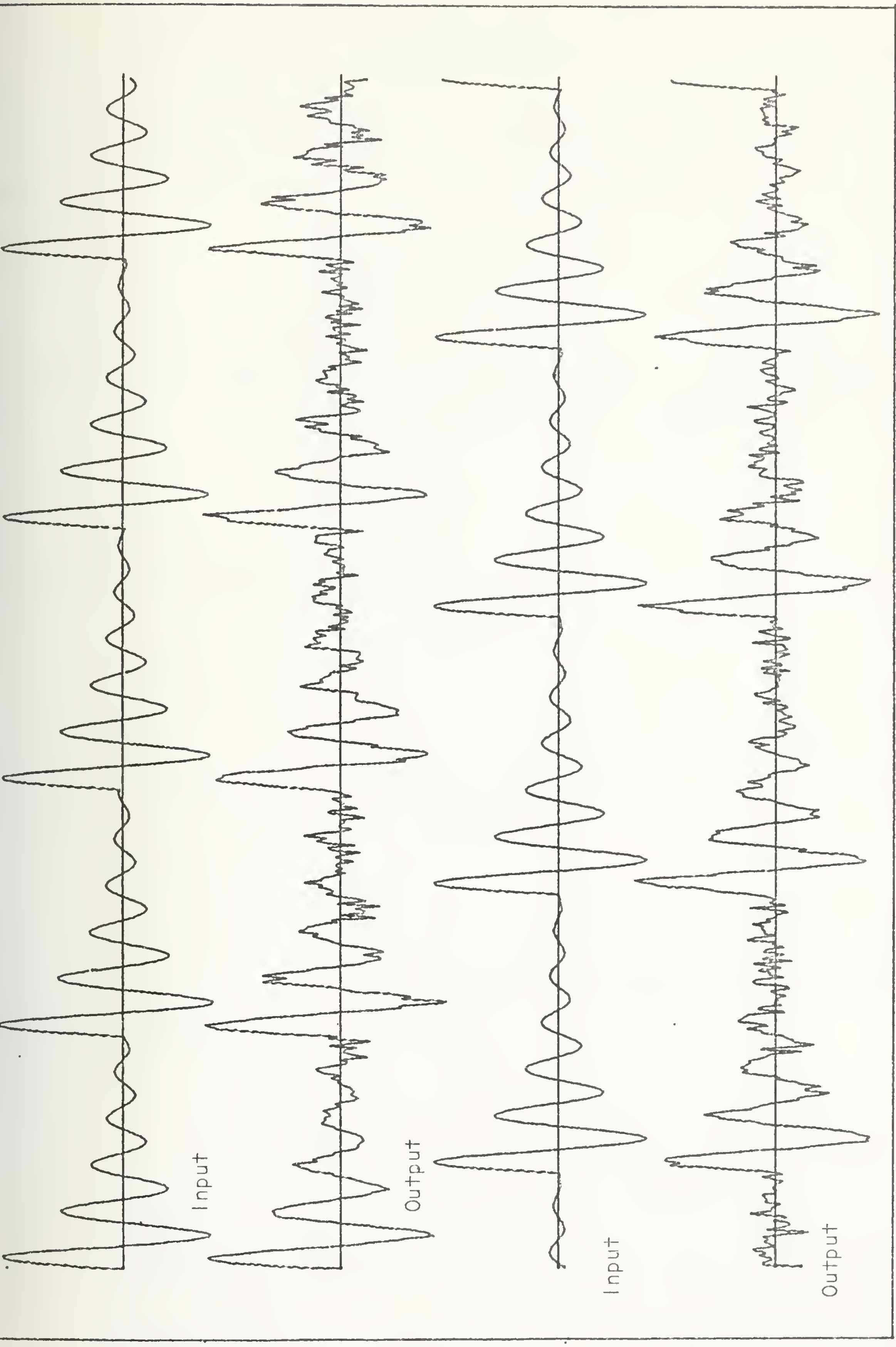


Figure 5-26 Input (noiseless) and Output Waveforms - Adaptive System, Seven Coefficients; White Noise Test

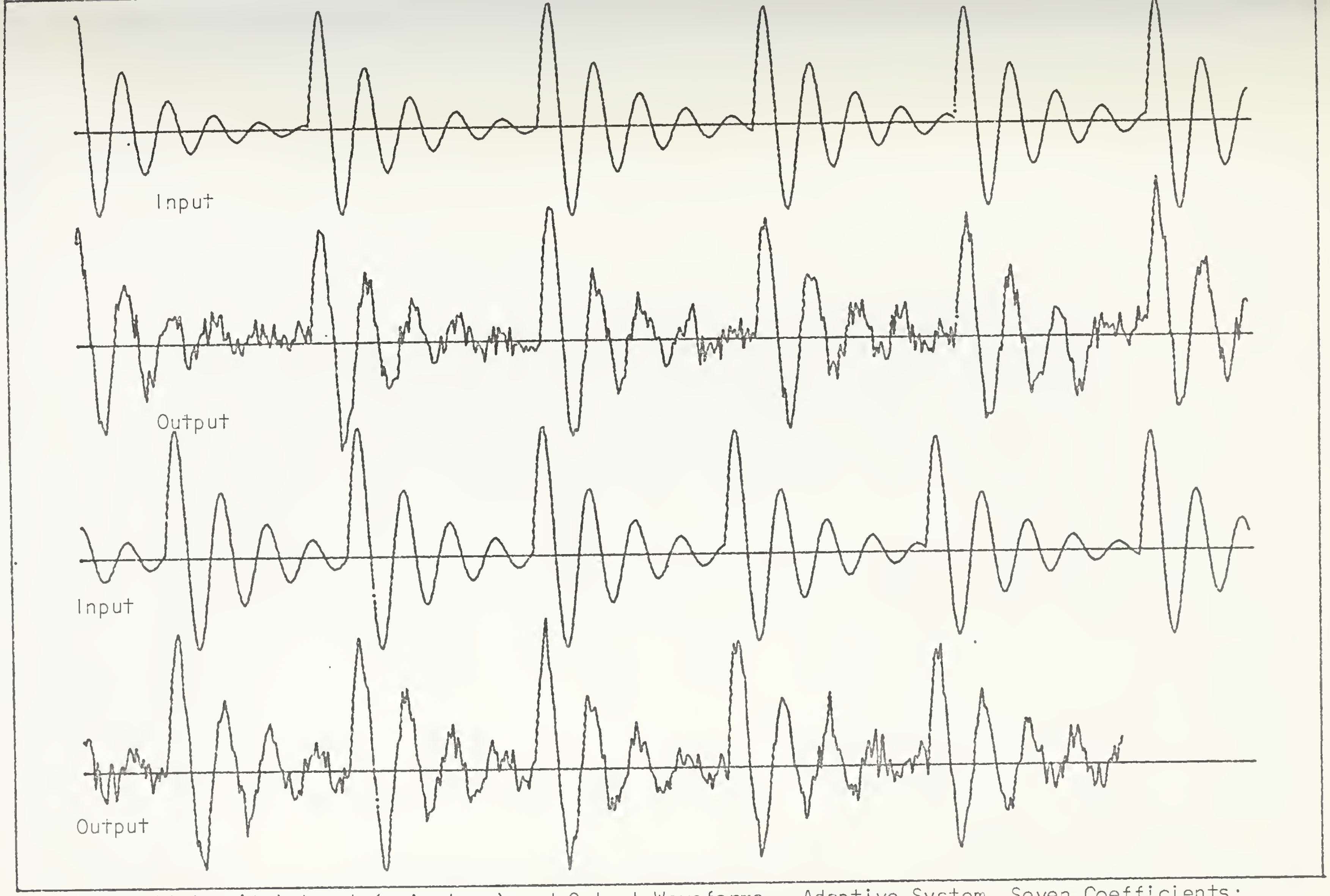


Figure 5-26 (con't.) Input (noiseless) and Output Waveforms - Adaptive System, Seven Coefficients; White Noise Test

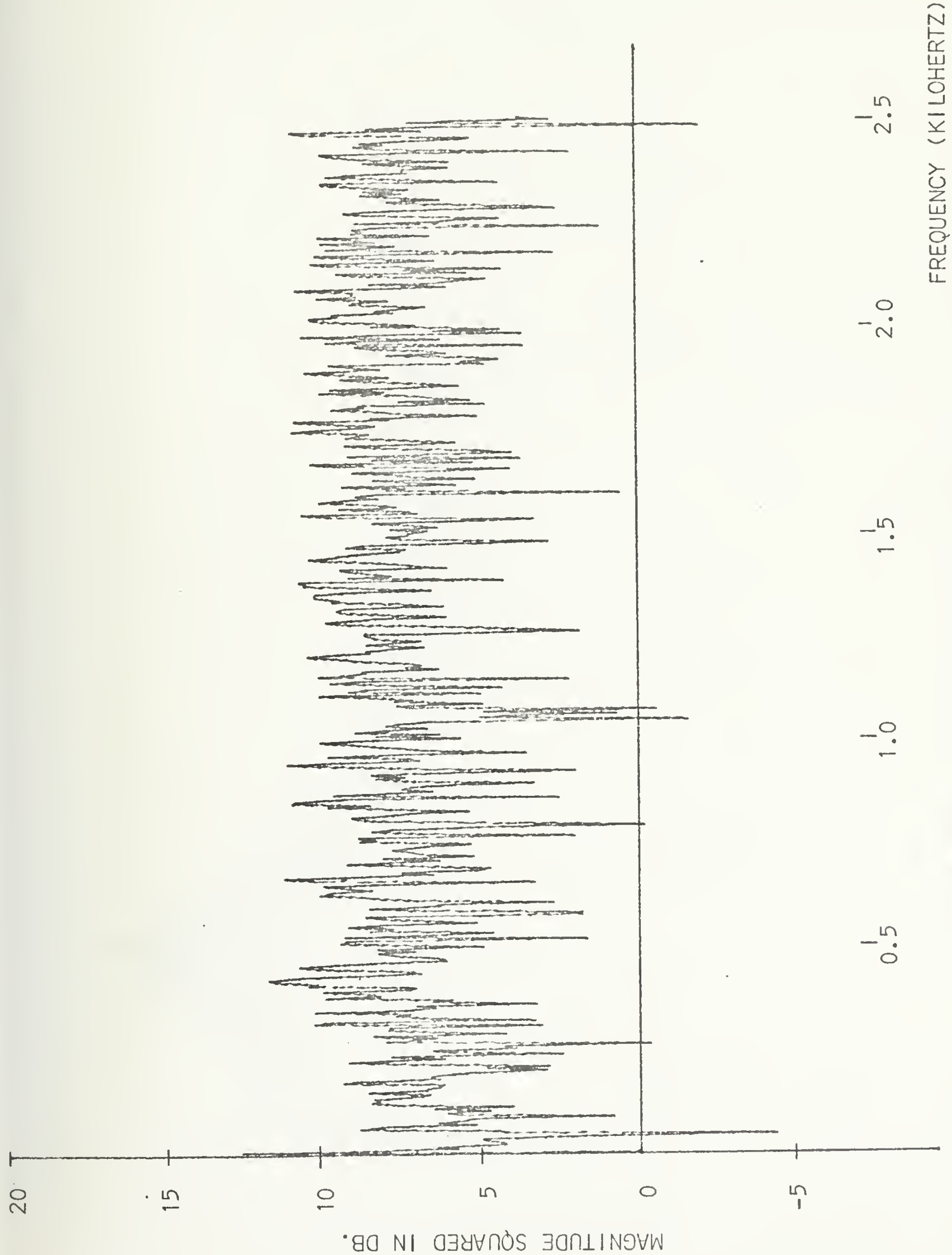


Figure 5-27 Spectrum of Additive Noise

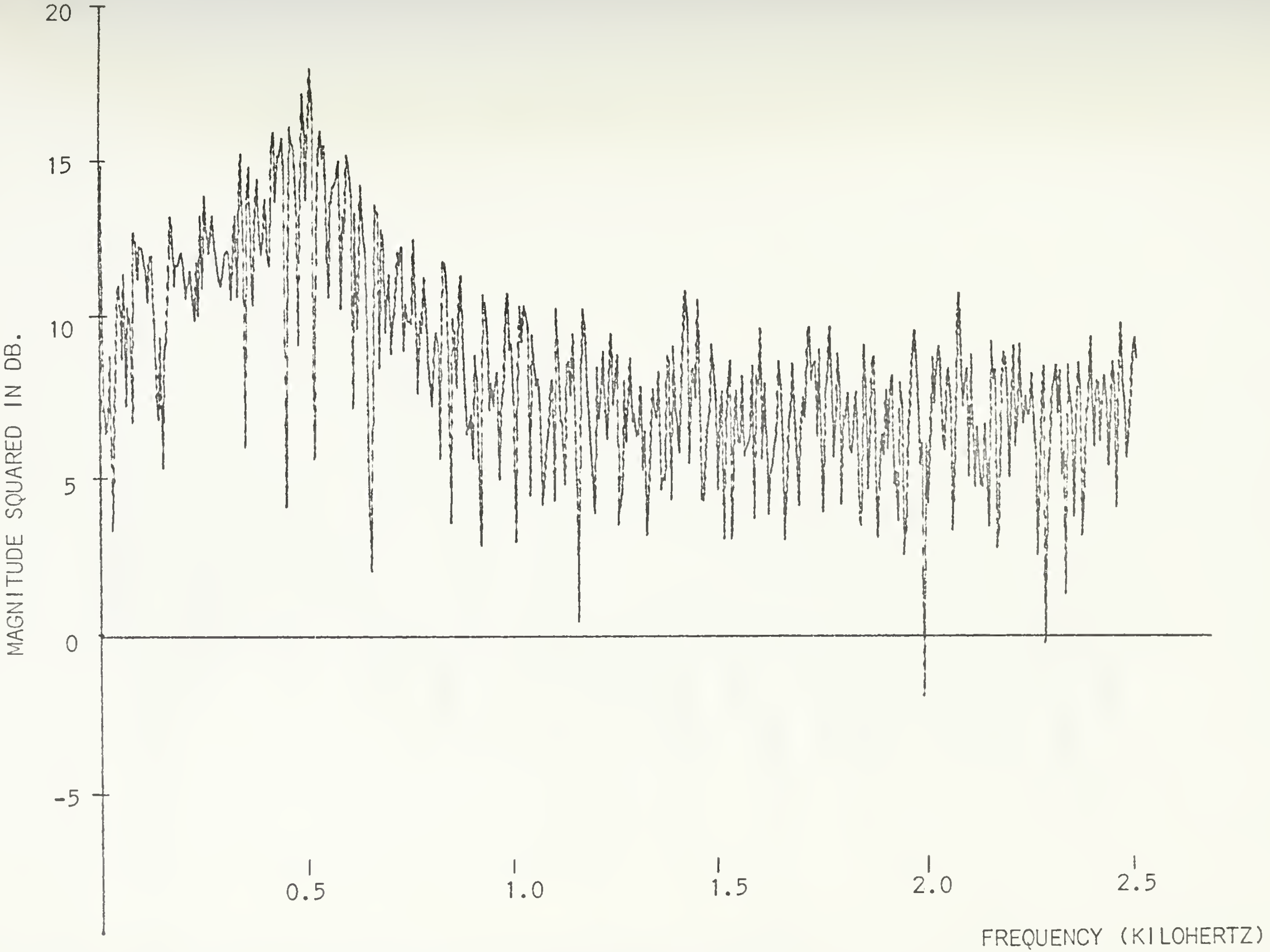


Figure 5-28 Spectrum of Input Signal plus Noise

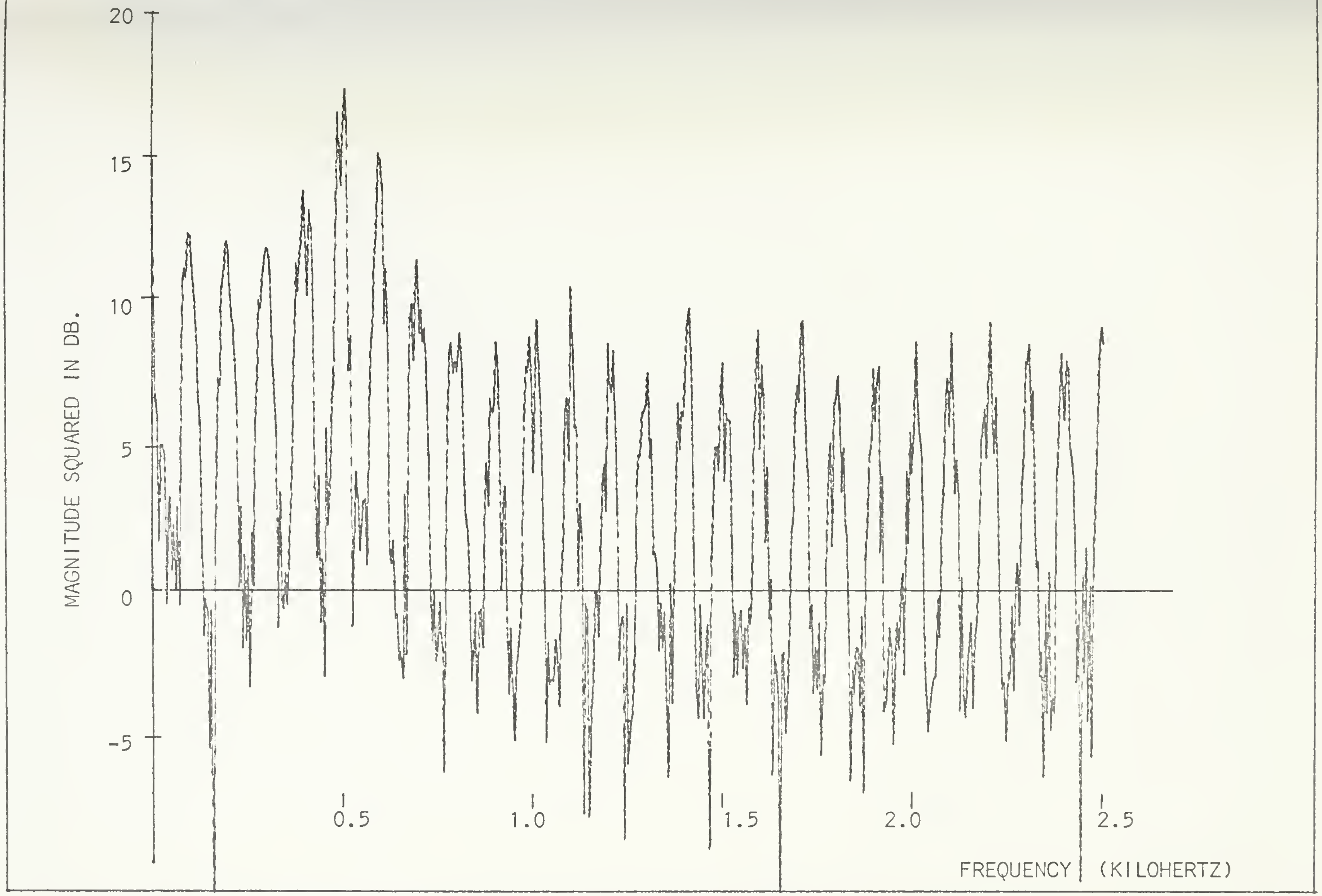


Figure 5-29 Output Spectrum - Shields' System; White Noise Test

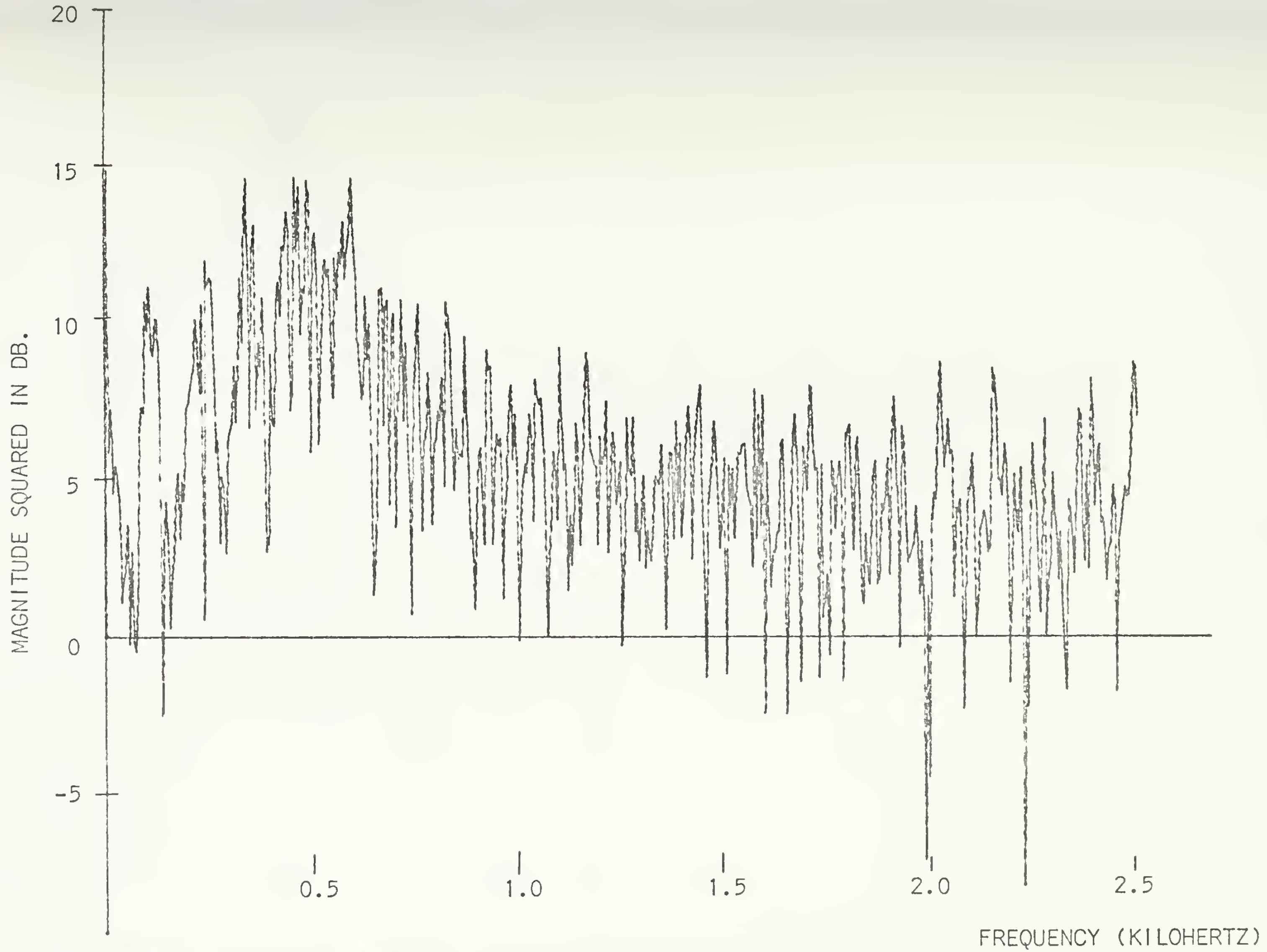


Figure 5-30 Output Spectrum - Adaptive Overload System; White Noise Test

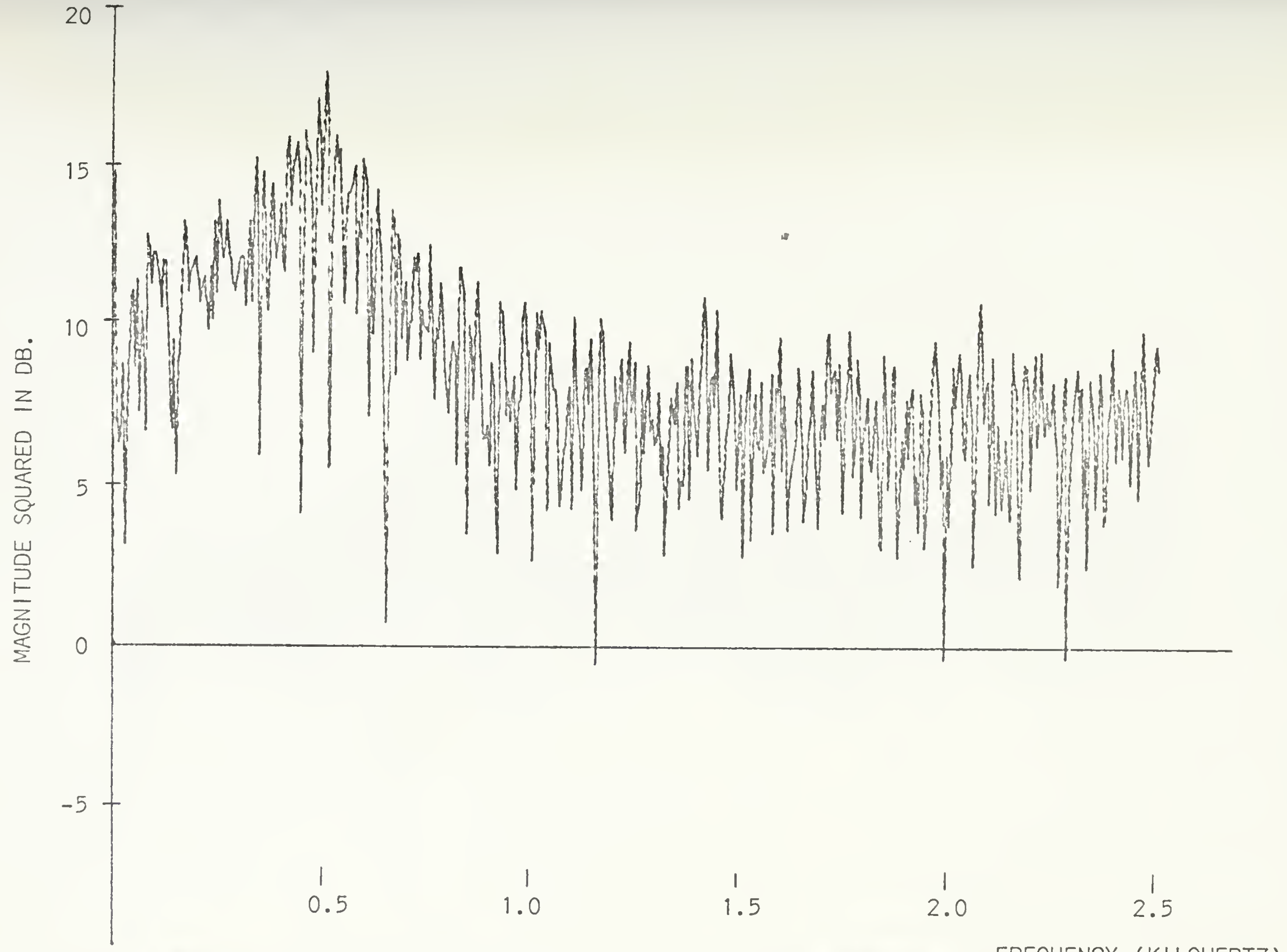


Figure 5-31 Output Spectrum - Adaptive System; White Noise Test

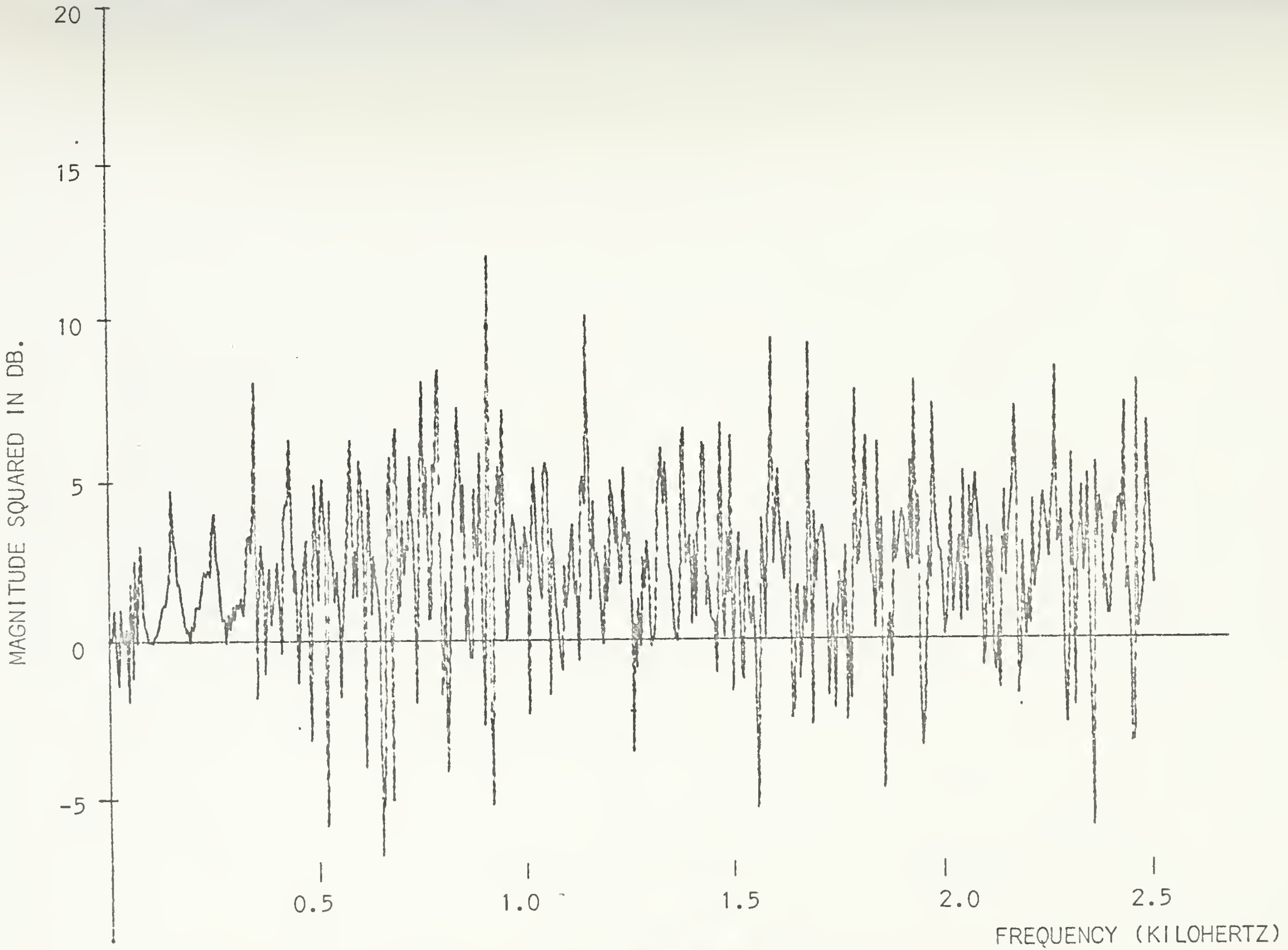


Figure 5-32 (a) Error Function - Shields' System; White Noise Test

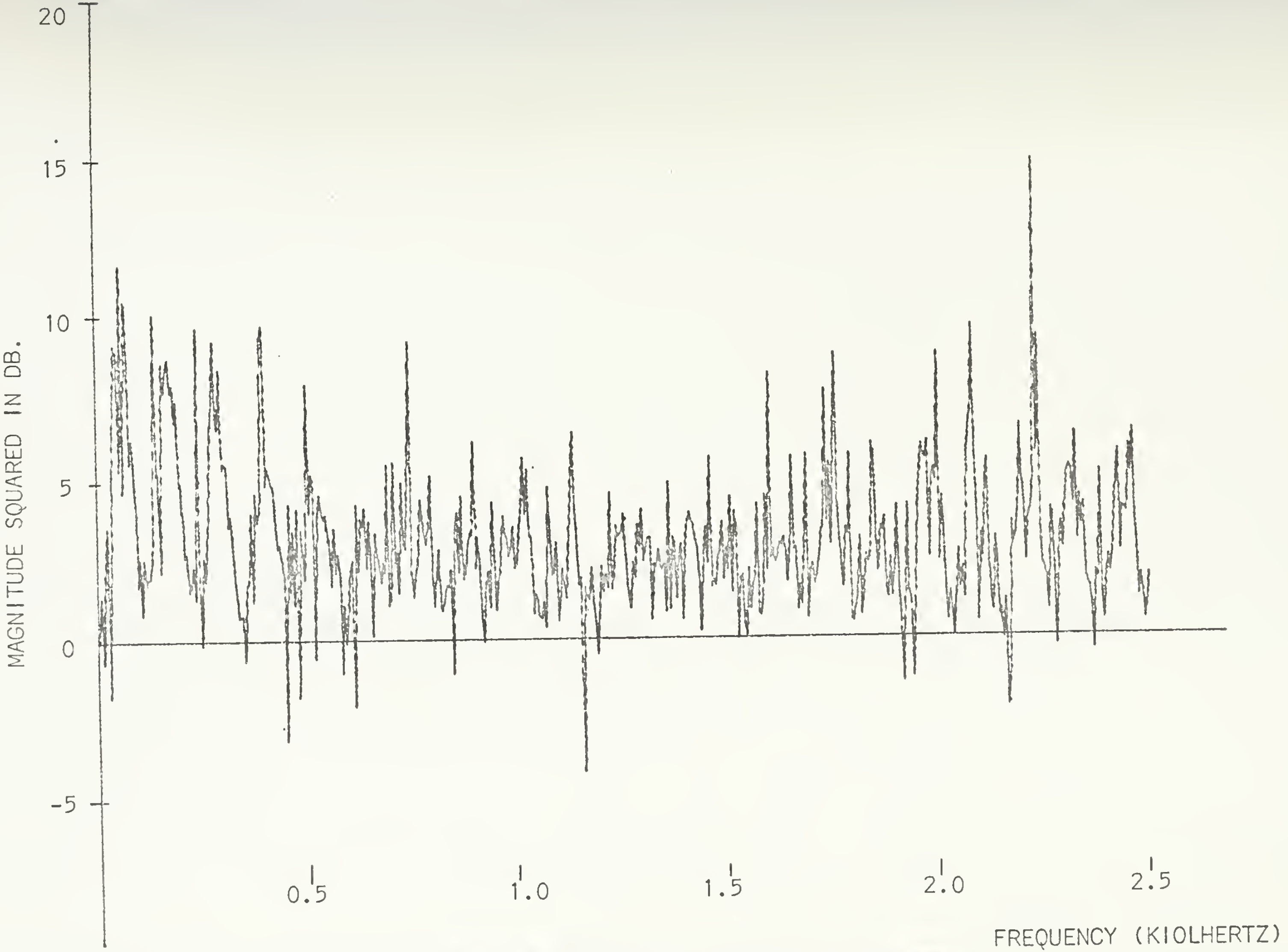


Figure 5-32 (b) Error Function - Adaptive Overload System; White Noise Test

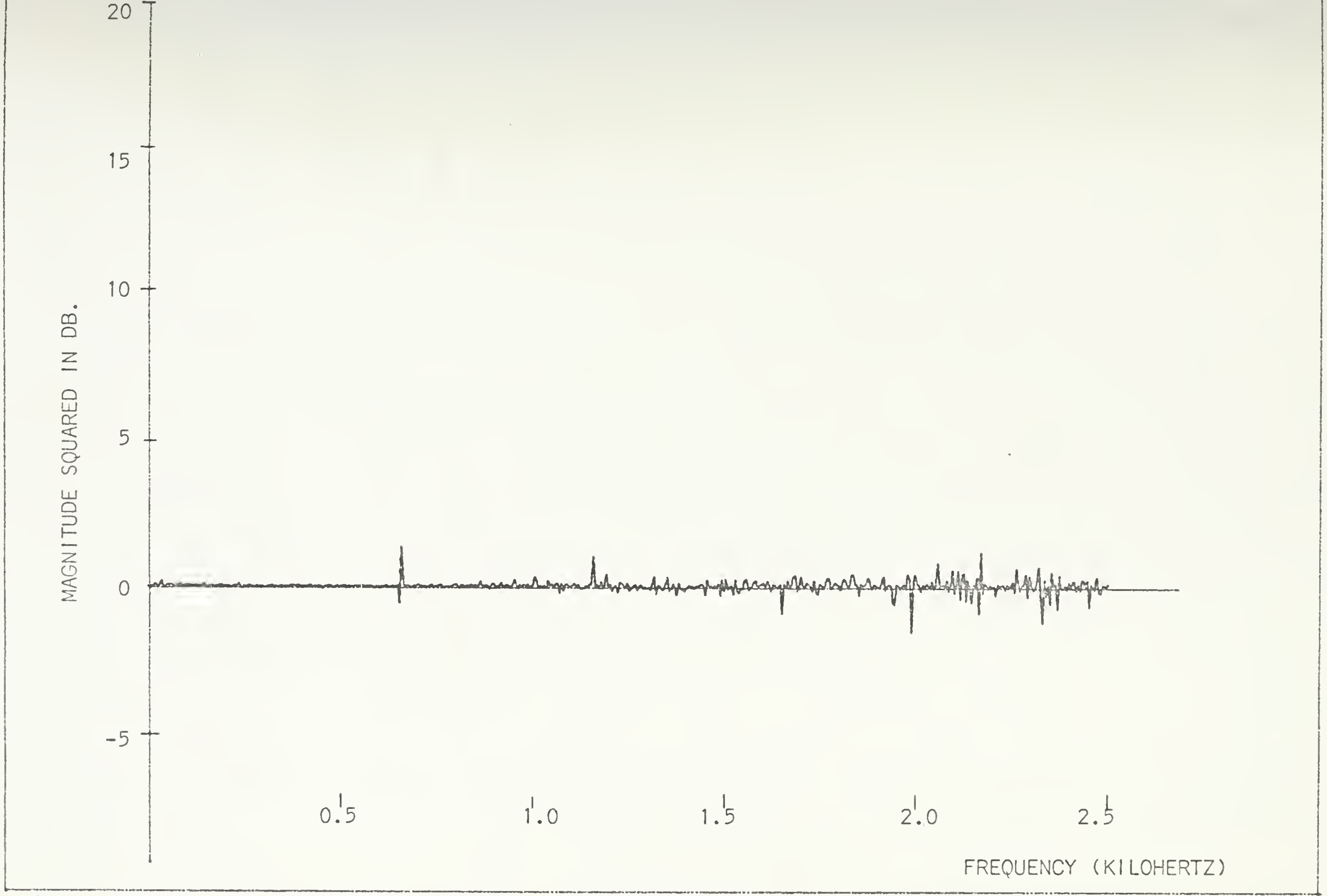


Figure 5-32 (c) Error Function - Adaptive System; White Noise Test

the other two methods provide an output signal that is both noisy and distorted.

The final test was conducted for the purposes of continuity. This test, as pointed out before, is a combination of tests one and two. The system proposed by Shields is presented in a time domain representation in figure 5-33. The adaptive overload and adaptive systems are shown in figures 5-34 and 5-35 respectively. The spectra of the three systems are shown in figures 5-36, 5-37, and 5-38 for Shields' system, the adaptive overload, and the adaptive systems respectively. Upon careful examination of these spectra, the level of the spectrum from Shields' system is generally lower at most frequencies than either of the adaptive systems or the spectrum of the desired speaker shown in figure 5-11. This point seems to agree with the phenomenon observed earlier in the input pass test. The amplitude of the output from Shields' system in the input pass test was being amplitude modulated, therefore, the waveform would have to contain less energy at some frequencies than an unmodulated waveform of the same type. The error functions shown earlier have been omitted for this test because they are almost identical and not very benefical for comparisons.

In summary, this series of tests was performed in order to provide some other means of system evaluation. It should be stated that the fact the adaptive filter did not suppress the noise waveform better than the system proposed by Shields is not surprising. The main advantage of the adaptive system is fidelity, not rejection. Some of

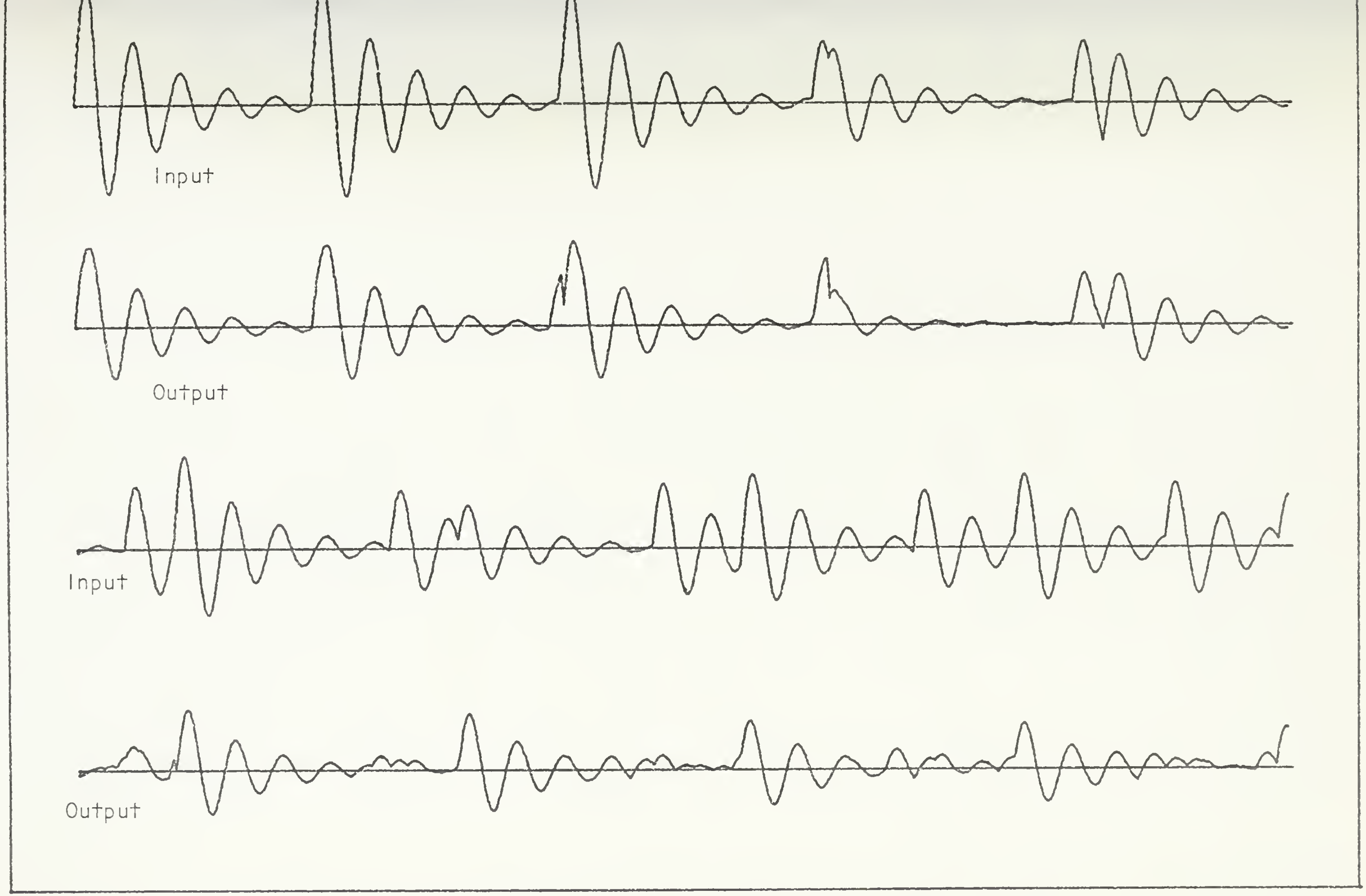


Figure 5-33 Input and Output Waveforms - Shields' System; Combined Test

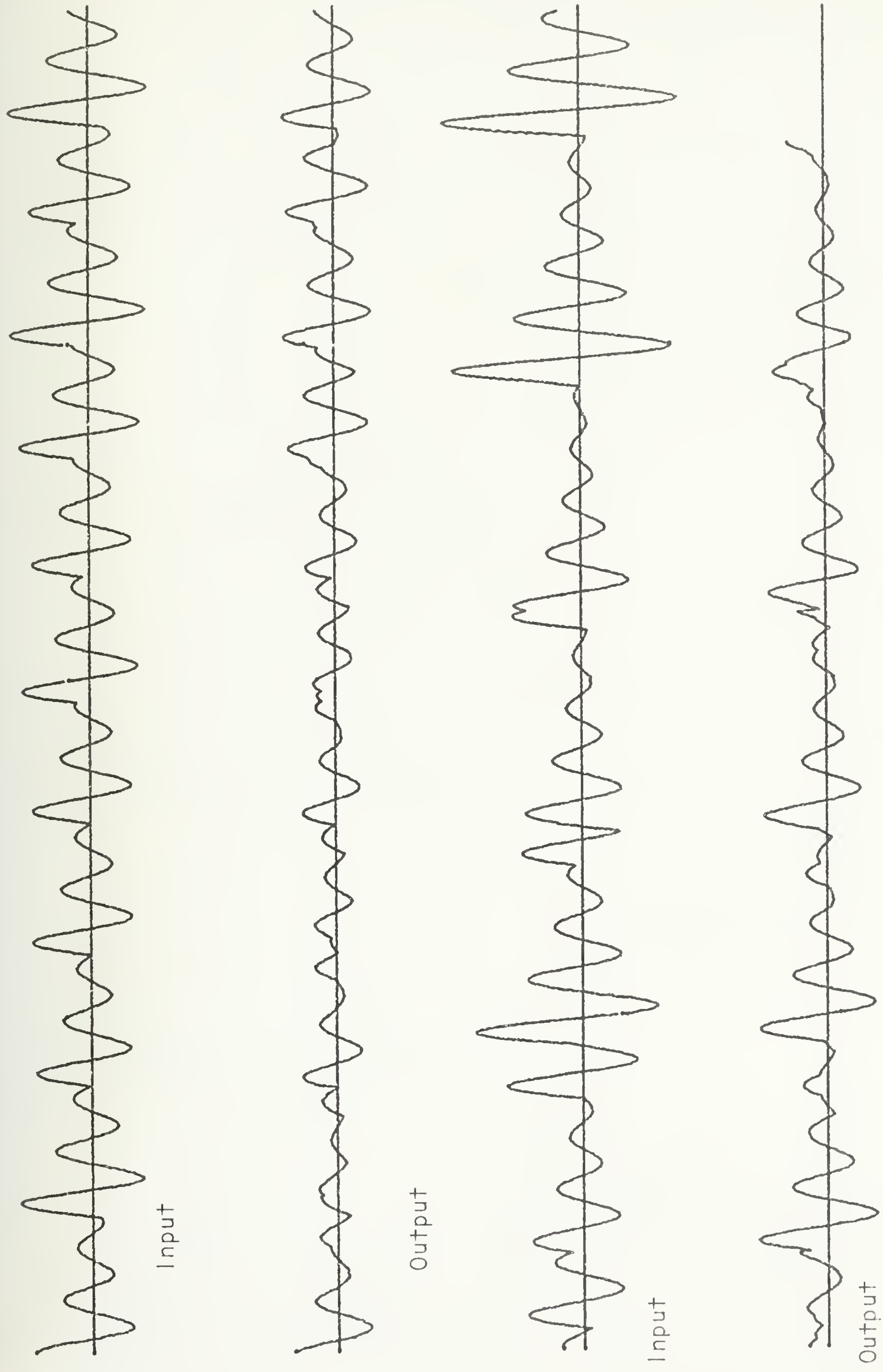


Figure 5-33 (con't.) Input and Output Waveforms - Shields' System; Combined Test

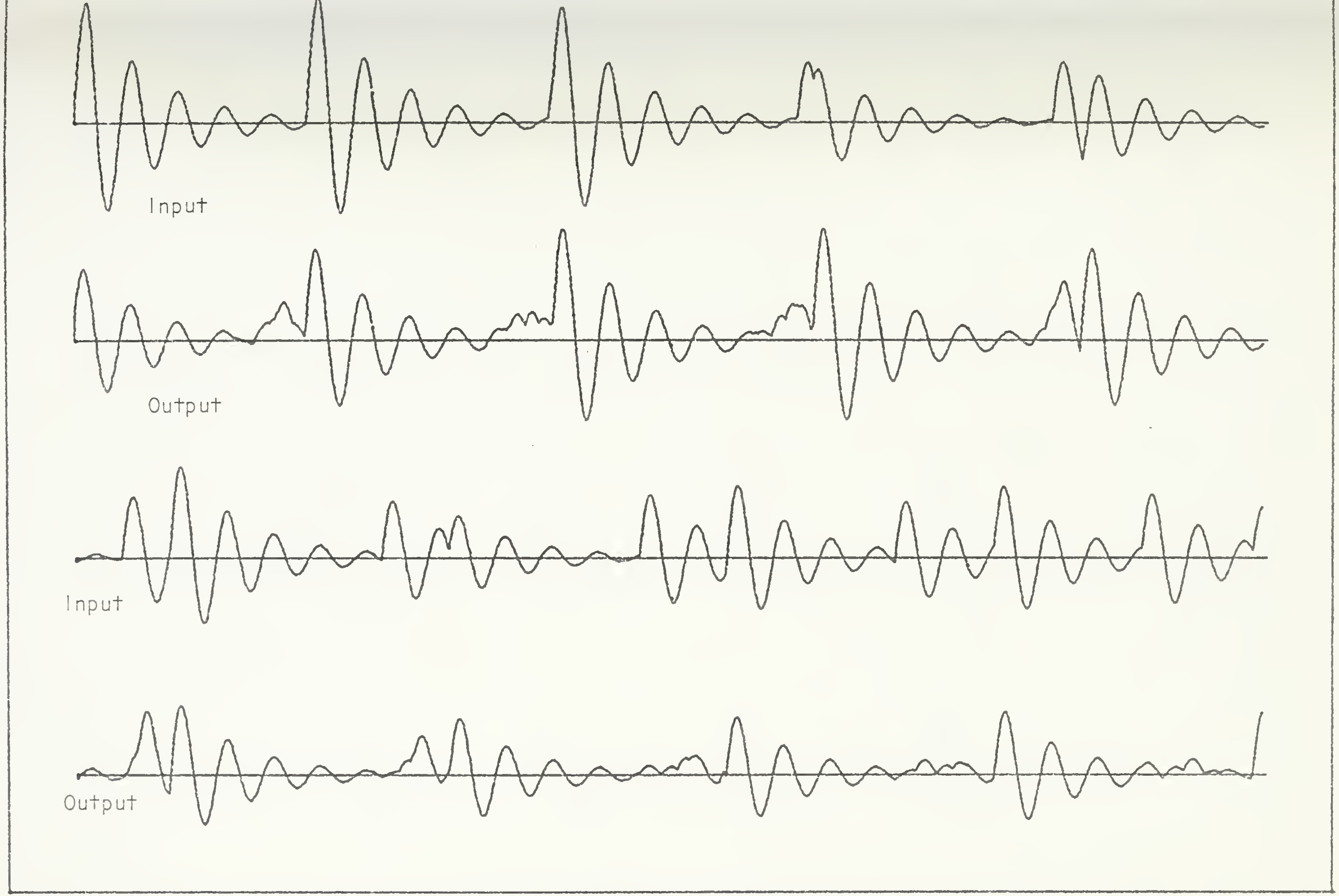


Figure 5-34 Input and Output Waveforms - Adaptive Overload System; Combined Test

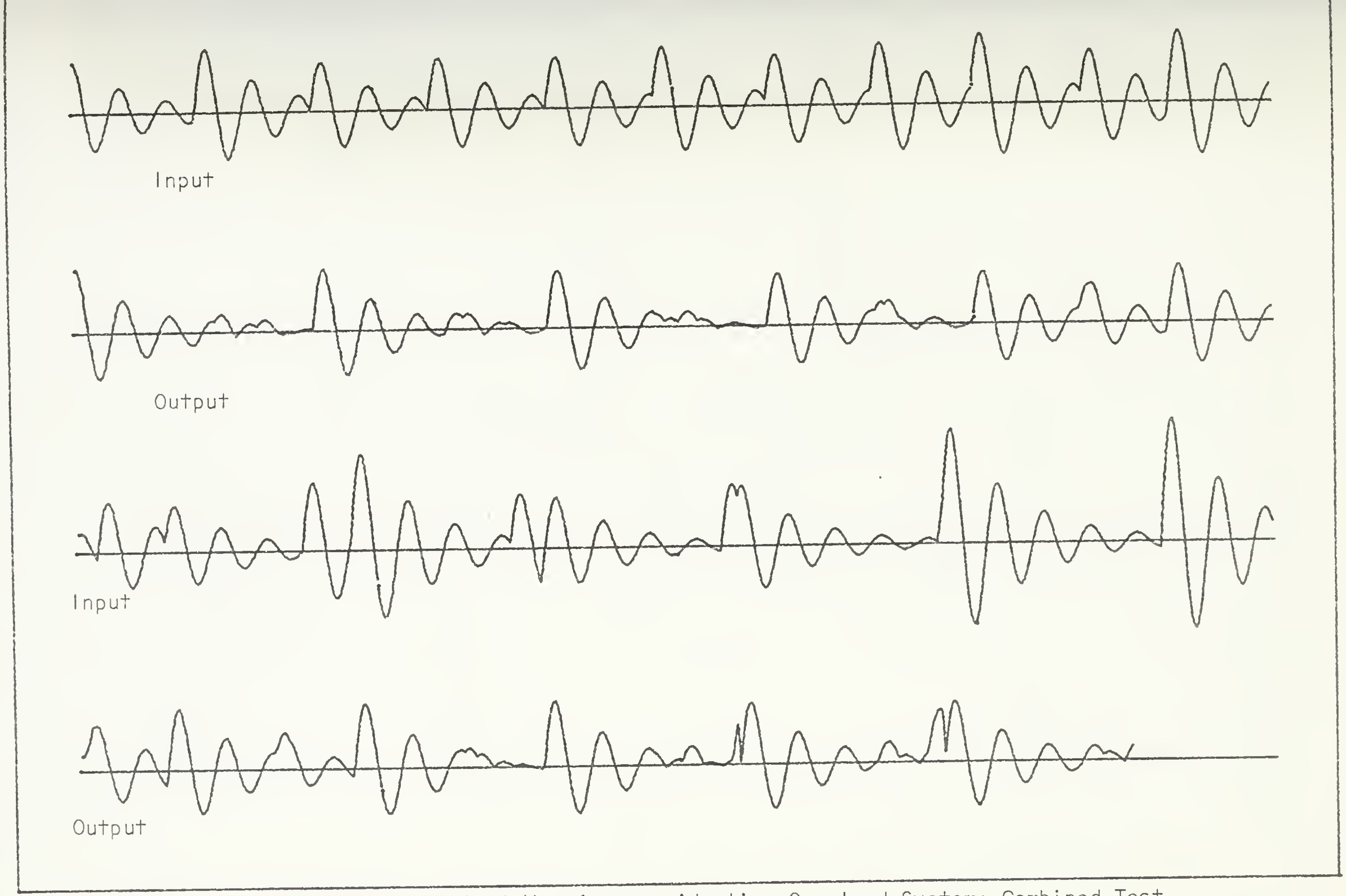


Figure 5-34 (con't.) Input and Output Waveforms - Adaptive Overload System; Combined Test

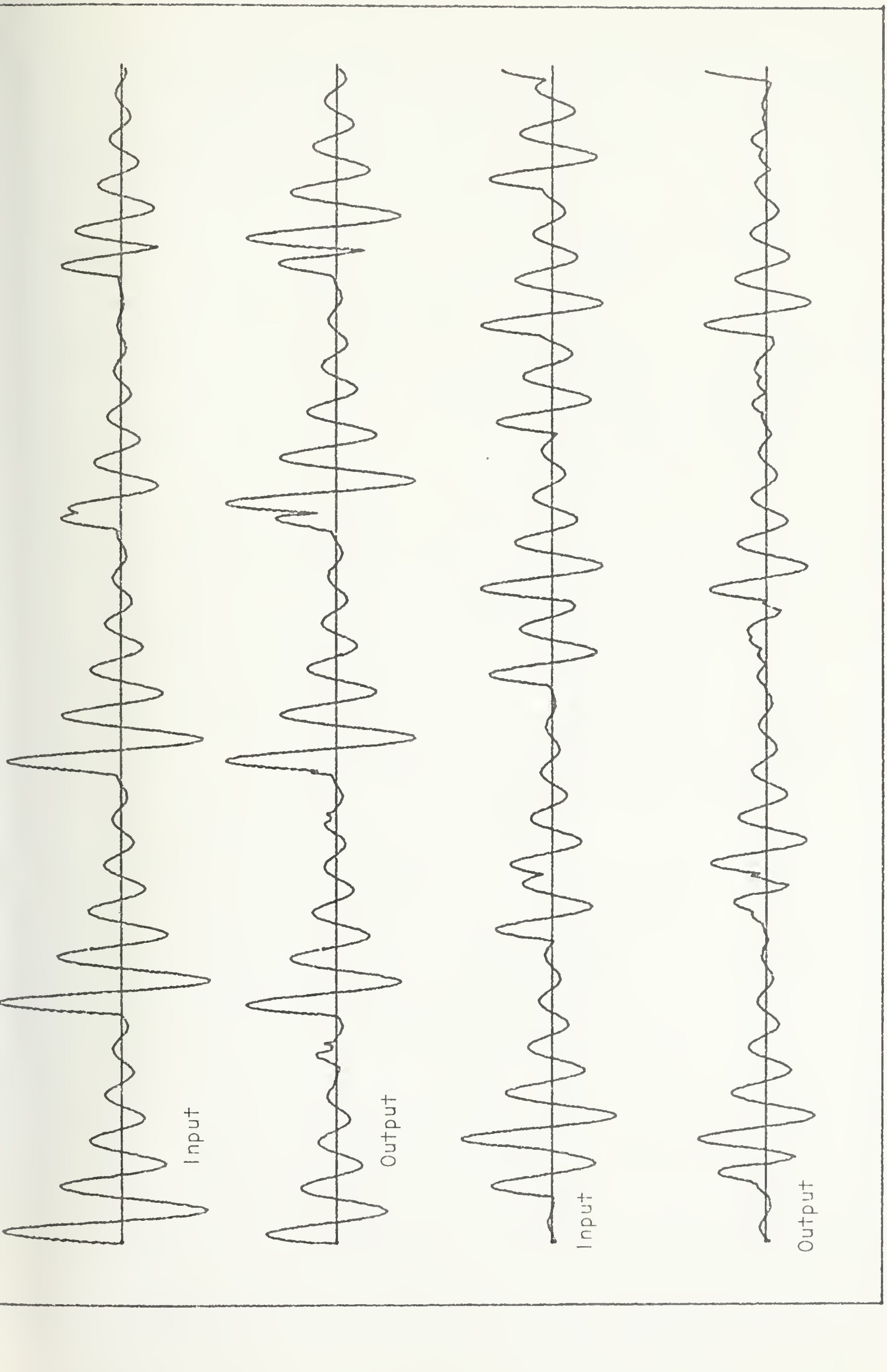


Figure 5-35 Input and Output Waveforms - Adaptive System; Combined Test

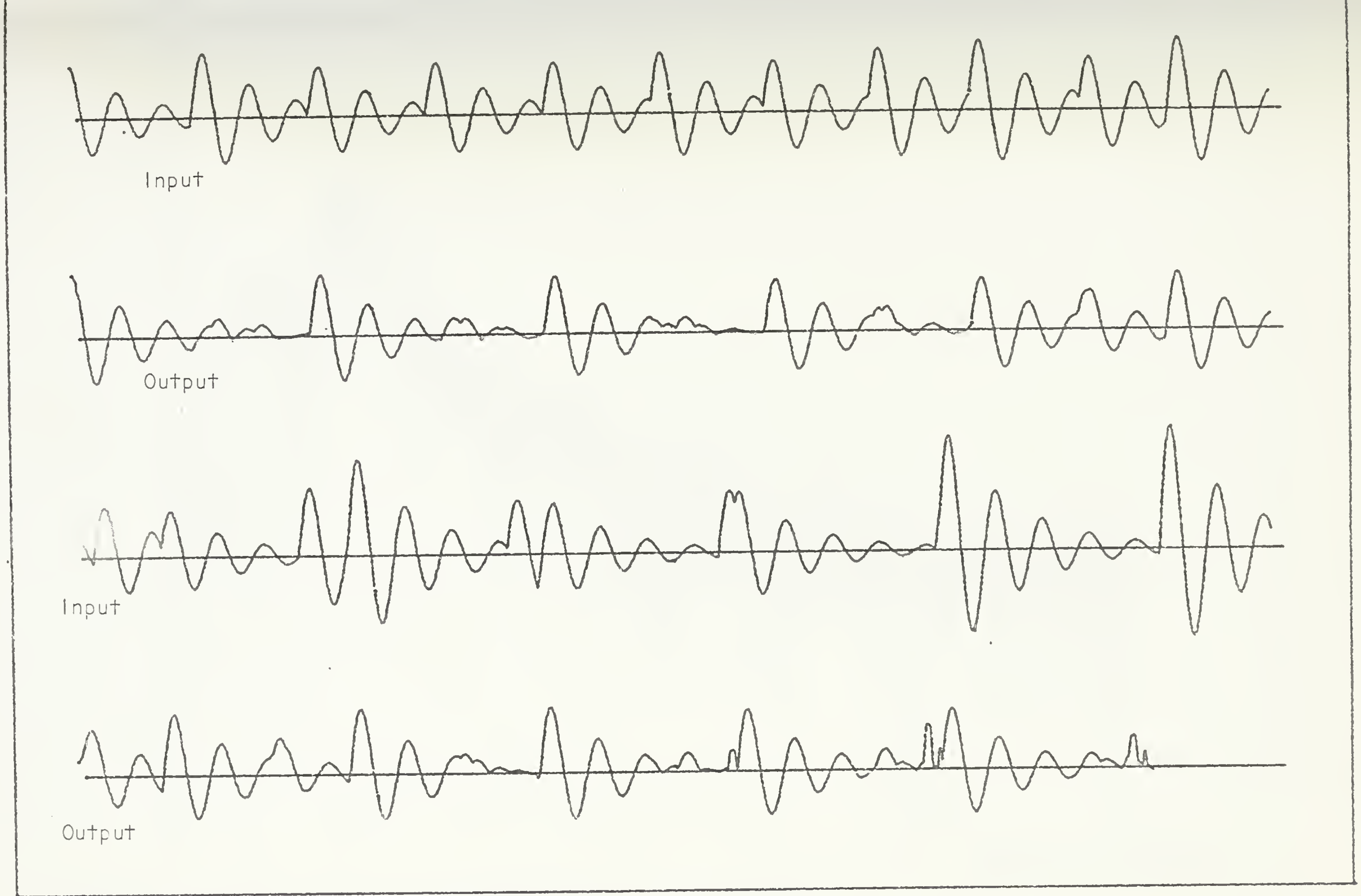


Figure 5-35 (con't.) Input and Output Waveforms - Adaptive System; Combined Test

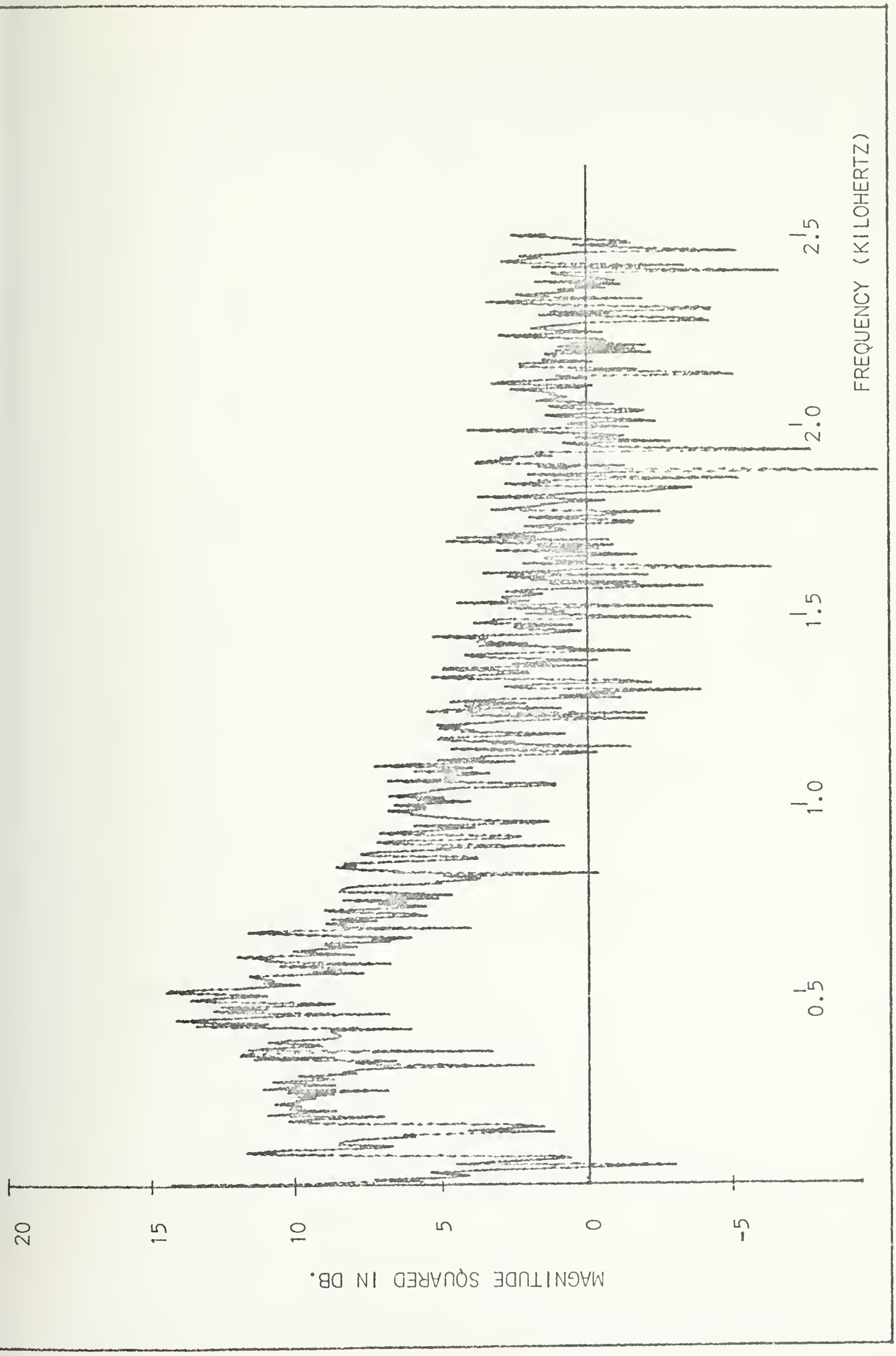


Figure 5-36 Output Spectrum - Shields' System; Combined Test

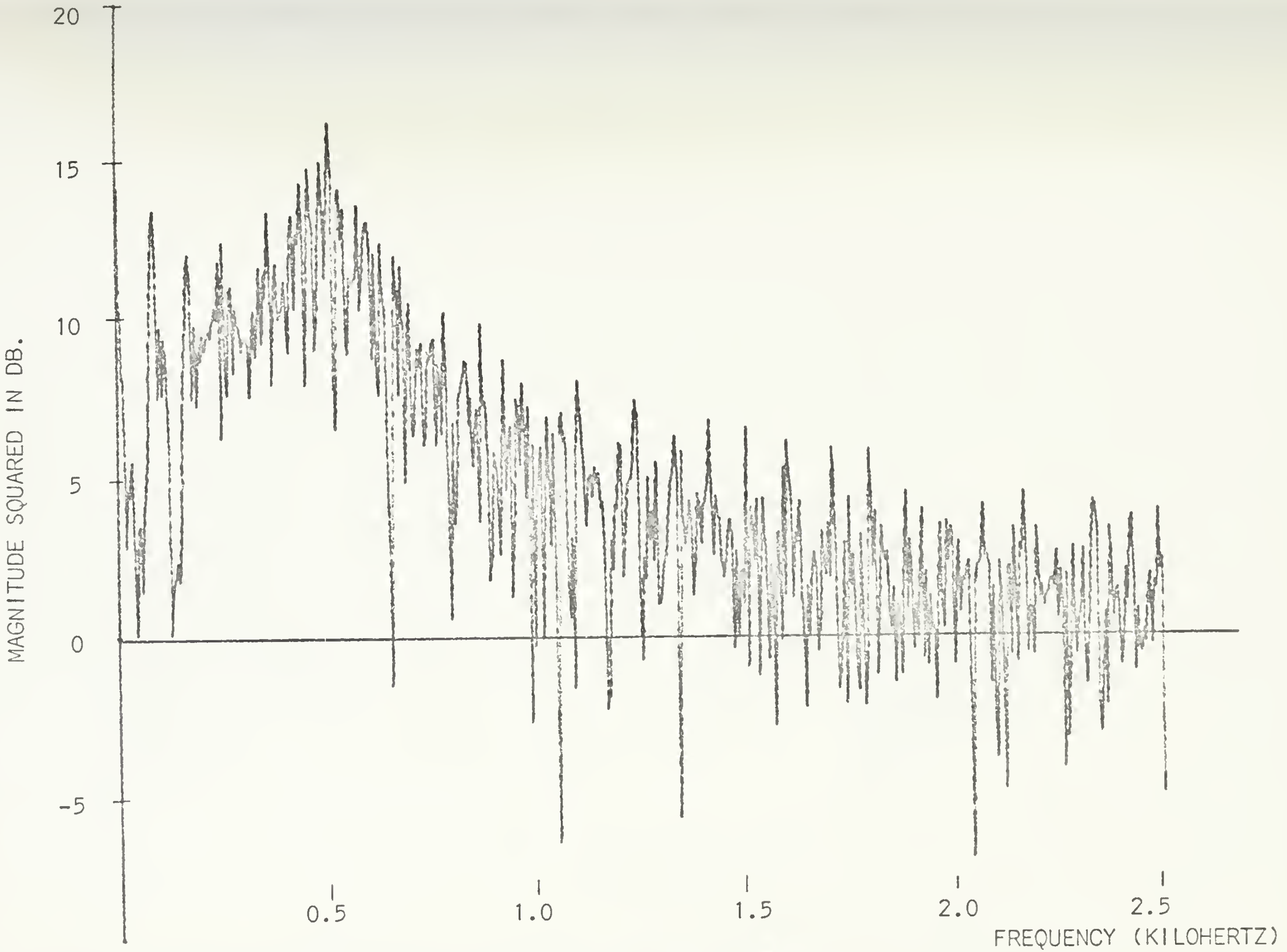


Figure 5-37 Output Spectrum - Adaptive Overload System; Combined Test

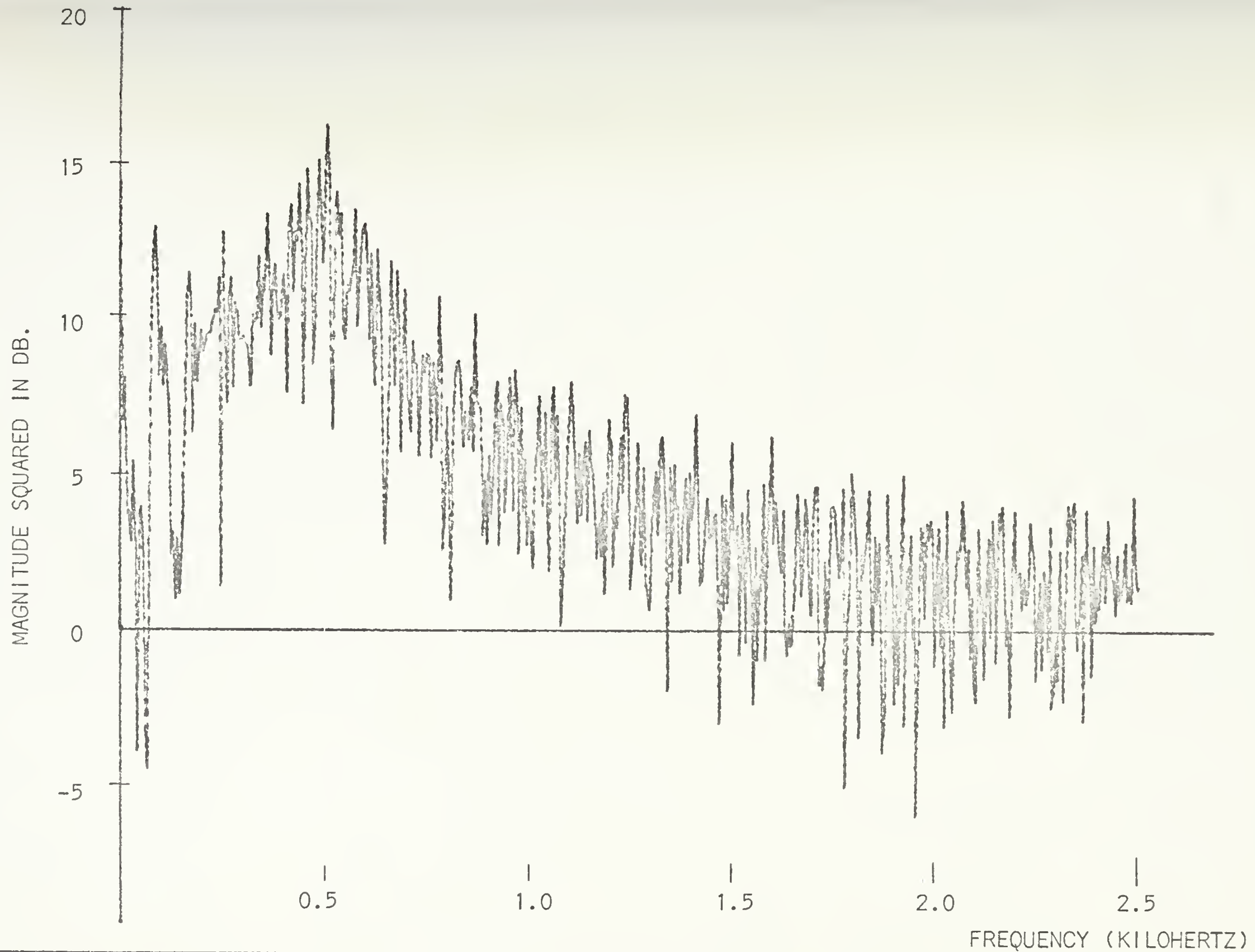


Figure 5-38 Output Spectrum - Adaptive System; Combined Test

the phenomena and errors observed from these test signals were helpful in implementing the systems that worked with actual speech. When the listening tests are performed, a more accurate comparison can probably be presented. However, before these tests are performed, other than informal listening evaluations and speech spectrograms, these results are all that are available for system comparison.

CHAPTER VI

COMPUTER IMPLEMENTATION

6.1 Description of Computer System

The computer work for this thesis was done on a PDP-11/45 digital computer. This computer is a highly sophisticated and powerful 16-bit word machine with 32 K of core memory and dual cartridge disk drives capable of holding 1.2 million words each.

Other peripheral equipment included the Lab Peripheral System, or LPS, a storage oscilloscope, a Hewlett Packard 7004 B plotter, and a VT05 Alphanumeric Display Terminal. The system was also equiped with analog lowpass filters, attenuators, and audio amplifiers.

The LPS is a modular, real-time subsystem that houses a 12-bit analog-to-digital converter (A/D), a programmable real time clock, and a display controller which includes two 12-bit digital-to-analog converters (D/A). With the LPS the task of input and output to the computer was easily handled.

The LPS real time clock was programmed to sample an input signal at 10 kilohertz after a lowpass filter with its cutoff frequency at 4.7 kilohertz. The computer was set to sample two channels simultaneously, and the speech signal along with the glottal accelerometer signal were put onto the disk in the form of a data file. The samples

were stored in an interleaving format so that the time reference between the two signals would be preserved. The program that conducted the sampling was set to sample up to 3.2 seconds of speech with its corresponding accelerometer signal. This amount of time was chosen so that most sentences could be accommodated, but with minor modifications, the program could be changed to allow any length file. The program that conducted the sampling also took advantage of the direct memory access (DMA) option of the A/D converter, which allowed the conversions to be stored in memory at the maximum rates without processor intervention.

The LPS also provided a capable display control. Along with the knobs and switches provided, a display program was written that was capable of displaying and editing waveforms. This feature was very beneficial in correcting pitch markings and in viewing processed and unprocessed waveforms. X and Y cursors with knob controls were included with an LED readout of position in order to locate specific points in the waveform.

The routine used for audio output was driven from the D/A converter in the LPS. The signal from the D/A converter was lowpass filtered at 4.7 kilohertz and then amplified. At this point the output signal could be recorded on magnetic recording tape for future use.

6.2 Shields' System

The filtering methods discussed by Shields were implemented on the PDP-11/45 using the standard RT-11 Fortran. It was decided that an

implementation in fortran would be a slower version, but after algorithms were perfected, the execution speed could be improved. Flowcharts for the systems described in this section are included in the appendix. Since the A/D and D/A converters used 12-bits, the input and output data values were in integer form although the internal multiplications were done in floating point format.

There were two different systems implemented for the comb filtering techniques, and the difference in these two methods was the manner in which the unvoiced segments were handled. A brief statement about these two methods seems necessary before the systems are described. The first method, the attenuated input method, stopped the comb filtering when an unvoiced segment was encountered. The program then began to attenuate the input by some specified constant until the next voiced section appeared. The second method, the "inertial filter", ignored the fact that an unvoiced segment had been encountered and continued to filter the input using the parameters of the last known pitch period value. The name inertial filter resulted from the physical aspects of the processing. The filter could be thought of as moving through the input samples with some velocity, and without any external forces, the filter and its motion would remain unchanged as it moved through an unvoiced section. With the exception of the manner in which unvoiced speech sections were handled, the two systems were basically the same.

The filter coefficients could be derived from several sources. A subroutine was written in order to let the source for the

coefficients and the number of coefficients be determined as the program commenced execution. The choices of sources included the four window functions that were used in the original system proposed by Shields. The window functions were the rectangular, the Hanning, the Hamming, and the Blackman. A second choice was the coefficients from a lowpass filter designed by the Parks-McClellan Algorithm that was available on the computer. The final choice was one that allowed the user to type in any coefficients desired.

After the coefficients had been determined, the input buffer was initialized with the first section of data. As mentioned in the previous section, the waveform had been sampled using two channels. The signal was stored on one channel, and the pitch marks were placed on the second channel. This type of implementation could be considered wasteful from the standpoint of storage, but from the aspects of the display, the waveform along with pitch marks could be viewed very easily. As the filtering commenced the center coefficient checked each value on the second channel for the pitch marks, and when a mark was encountered, the filter was changed. A pitch table file contained the distance between pitch marks and was used to furnish a new pitch period value after a mark was encountered. If the value of the pitch period was greater than 20 milliseconds, the section was treated as an unvoiced section. The unvoiced speech procedures described above were then implemented depending on the system.

The pitch table was also marked to denote areas that were silent. These silent areas were detected as mentioned in Chapter III from an

energy measurement over a segment of data. These silence marks were used only to speed up the processing at the beginning and ending of a sentence. If a value from the pitch table indicated a silent area, the output of the filter was set to zero, and no multiplications were performed. Procedures for handling the silent areas that appeared inside the sentence limits were not thoroughly investigated, and this could be a topic for future research.

From all practical aspects and to the best of the author's knowledge, these systems were identical to the ones described by Shields. The execution time for these systems was on the order of seventy-five times real time.

6.3 Notch Filters

Another method that was investigated for the speech enhancement problem was a notch filter implementation. Shields suggested that a notch filter be used with the pitch marks of the unwanted speaker to reject the unwanted speaker.[33]

In the comb filter implementation, the impulse response of a low-pass function was used. Using the principle of duality a notch filter implementation yields a frequency response for one of the possible filters as shown in figure 6-1.

The high pass prototype filter was designed by means of the Parks-McClellan Algorithm. There were several parameters that had to be varied in order to decide what type of prototype filter was needed.

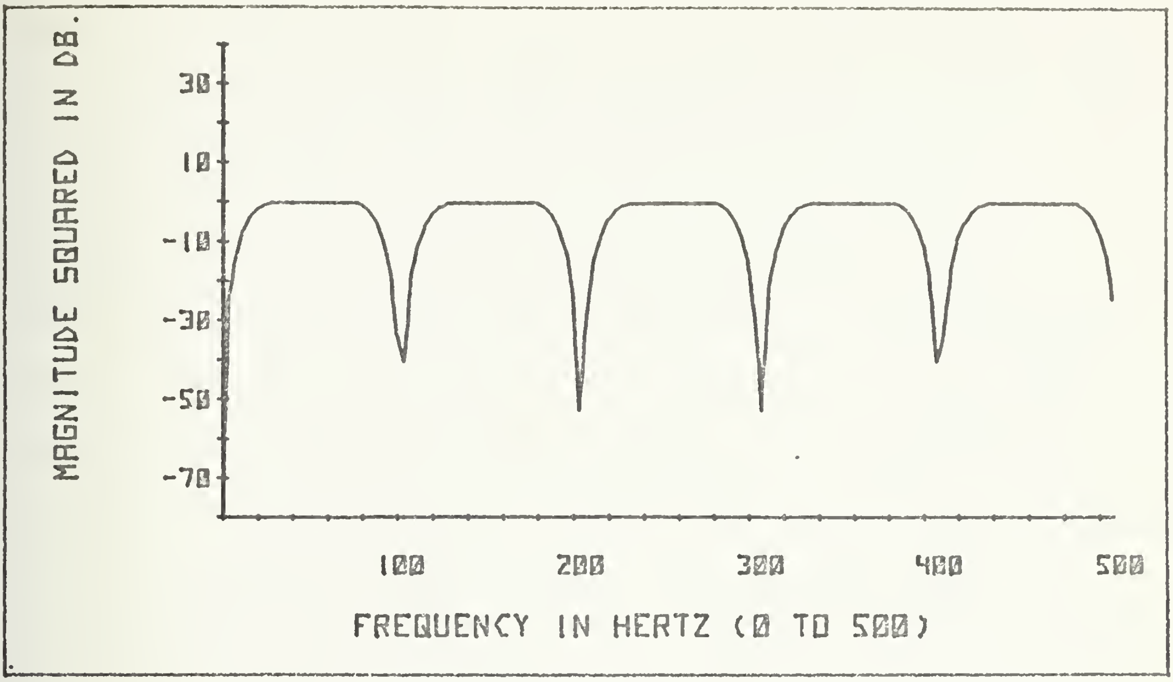


Figure 6-1

These parameters were:

1. Filter order
2. Stop Band Width
3. Transition Width

The filter order of the prototype high pass filter was limited to less than twelfth order. This value was determined from the implementation. If a number of zeros equal to the pitch period were inserted between coefficients, then the effective order of the notch filter grows quite rapidly. For example, if an eleventh order prototype filter were chosen and the pitch period were 100 hertz, then, the order of the notch filter would be 1100th order. A filter length of this magnitude is on the borderline of the quasi-periodic assumption

made for the speech waveform. For the most part, the high pass prototype filter order was between seventh and ninth order.

The second consideration was the width of the stop band in the prototype filter. This parameter can be linked directly to the width of the spectrum of the speech waveform at the various harmonics of the fundamental. If the stop band is too narrow, then the energy at each harmonic from the unwanted speaker will not be sufficiently suppressed. On the other hand, if the stop band is too wide, the notch filter may suppress the desired speaker and cause distortion. The above discussion relates back to the basic tradeoff for speech enhancement. Again, this tradeoff is the desired speaker distortion versus the undesired speaker attenuation.

The final primary consideration, the transition width, was related to the first two considerations already mentioned. The transition bandwidth can not be made sufficiently low without increasing the order of the filter. As the transition width is decreased without changing the filter order, the ripples in the pass and stop bands increase. Therefore, the limit for the transition width is related to the maximum ripple that can be tolerated.

From the prototype filters that were designed the transition widths were relatively large due to the limitations in the filter order.

After the coefficients from the prototype high pass filter had been calculated, these coefficients could be used in the same programs as used in the comb filtering implementation.

6.4 Adaptive System

The adaptive filtering systems were implemented much in the same manner as those systems mentioned in the previous section.

Since the methods used to implement the system that corrected for the overload problem encompassed the methods used for the system that did not, only the former system's description will be covered.

The initialization was the same as in the previous description, but an array was used to hold the information on the spacings between the coefficients since this system did not use uniformly spaced coefficients. Initially all spacings were set equal to the first voiced pitch period. As the filter moved through the input data, and pitch marks were detected, the spacing array was continually updated to reflect the correct spacing for each coefficient. If an unvoiced area were reached, the scheme was to perform the inverse operation described above. The coefficient that moved into the unvoiced area would retain the last voiced pitch period spacing, and the filter continued in this manner until all coefficients were clear of the voiced area. In unvoiced areas the input was attenuated as in the previous system.

Another array was used to correct for the overload problem. This array contained the initial starting locations in the buffer after the filter had been changed. If a coefficient were about to enter an area where overloading would occur, that coefficient was set to zero, and the rest of the coefficients were rescaled so that the output would

have a constant level. The array that contained the starting locations was used to detect when an overload situation existed.

The rest of the system involved only bookkeeping operations that were needed to handle the rules used by the system. The system was a little slower than the one proposed by Shields and operated on the order of ninety times real time.

In summary it should be pointed out that the fortran implementations were not designed for speed. Their basic design philosophy was one of user interaction with the programs to ensure correct processing. If speed had been a factor in this implementation, several items could have been changed. First, the input waveform could be composed of a single channel, and since the data samples were only 12-bits long, the higher order bits could be used for the pitch mark information. With this change in effect longer segments of data could be stored in core memory, eliminating several input/output operations. The filter implementation could take into account the even symmetry of the FIR filter and eliminate one half of the multiplies per output point. The addition of assembly language subroutines to do the calculational aspects of the program may also speed up the programming. With the implementation of these changes, the system begins to lose the user interaction capabilities that have proven to be very helpful in this work.

CHAPTER VII

SPEECH WAVEFORM RESULTS

In order to determine whether or not the systems would perform as planned on real speech waveforms, several sentences were processed by the various systems. A true evaluation of the performance of the three systems involves listening to the processed outputs when speech signals are used. These systems will be tested in extensive listening tests at a later time, but from informal listening results and spectrographic analysis will be discussed in this chapter. It should be pointed out that it is very difficult to describe the results of the various systems with speech waveform inputs. The descriptions and comparisons of the systems will be made with characteristic words that are not quantitative in the least.

The initial parameters used in processing the speech waveforms originated from the information provided in Shields' thesis. Shields concluded that the optimal value for the number of coefficients was seven. This number of coefficients provided the best compromise for the tradeoff between desired speaker distortion and undesired speaker separation. He stated that the Blackman Window provided the best set of coefficients from the window functions used, but he also stated that the differences in the results between the Blackman and Hamming Windows were slight.[34] With these facts in mind, these parameters

could remain fixed while the enhancement systems were varied.

Before the results of the tests similar to those in Chapter V are discussed, a few phenomena characteristic to the individual systems will be introduced.

7.1 Comb and Adaptive Filter Results with Speech Waveforms

There were several phenomena observed in the different systems that were not noticed in the test signal section. In the attenuated input method of Shields' system, an attenuation constant that was too small caused the output waveform to appear chopped. This chopping sensation resulted from unvoiced sections that were being attenuated by a great amount, and this caused a drastic change in the amplitude between a voiced and an unvoiced section. At attenuation constant of 0.3 seemed to be the smallest value for the constant that allowed the chopping to be imperceptible.

The inertial filter implementation of Shields' system did not produce the chopping sensation, but a soft reverberation was introduced which was audible in quiet. This reverberation could be directly attributed to the fact that unvoiced sections were processed by the comb filter with constant parameters. Both the chopping sensation and the soft reverberation were phenomena that could be attributed to the unvoiced speech segments. Although the time intervals for unvoiced sections in normal speech are small, the phenomena occurring in these unvoiced sections seemed to carry over into the voiced sections and disguise the actual results for the voiced sections.

The same types of tests performed on the test signals in Chapter V were implemented on the actual speech waveforms. These tests were evaluated by informal listening and by spectrograms. Spectrograms will be included as a final comparison on the methods that were implemented. In these tests the two systems implemented for Shields' method were equivalent since the desired speaker consisted of a totally voiced waveform.

For the input pass test all systems performed equally well for the most part. The most noticeable characteristic of Shields' system was a slight hoarseness in several words. The adaptive systems were characterized by a slight reverberation. All output waveforms had very similar spectrograms with only very minor differences from the spectrogram of the input sentence.

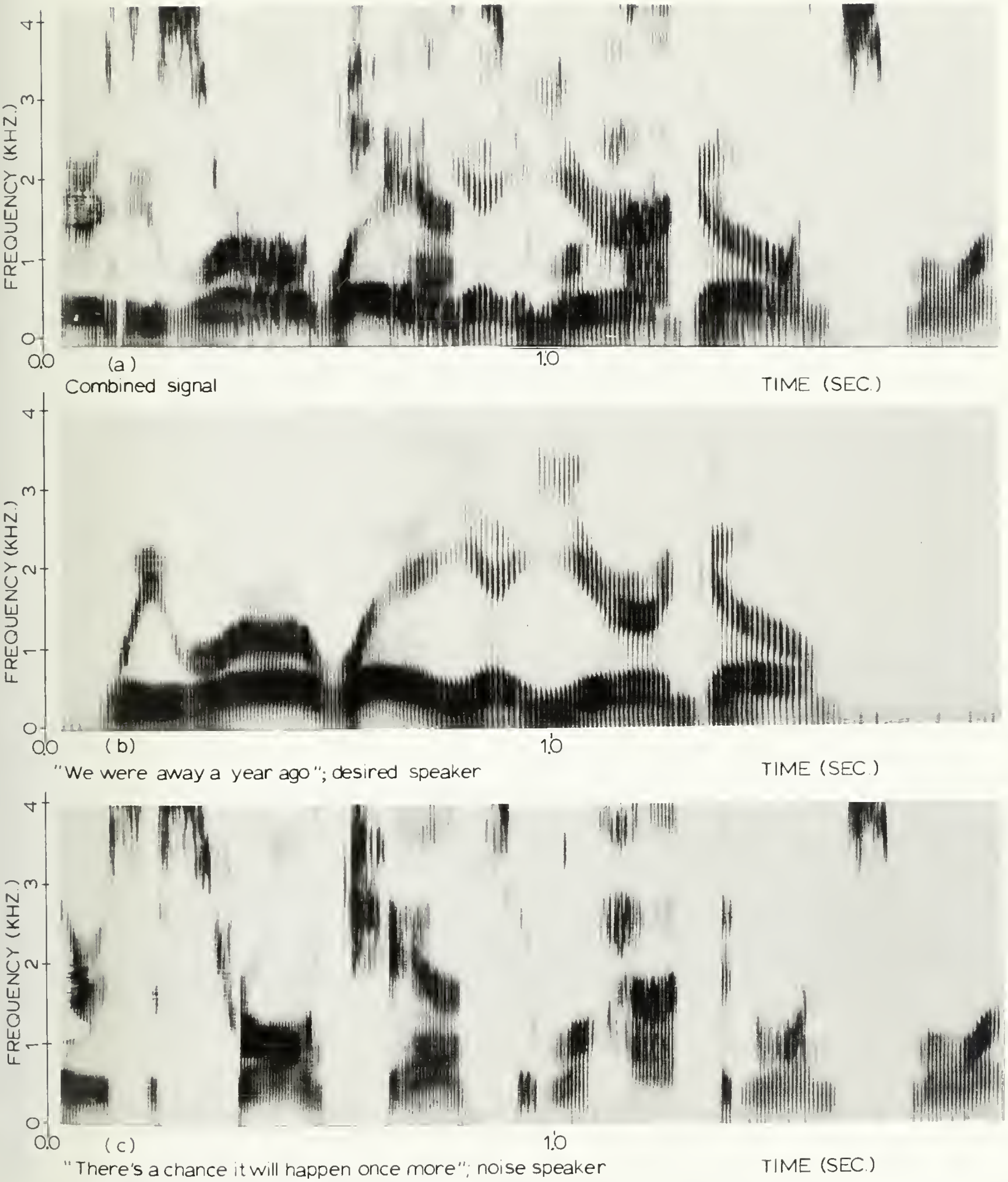
The input reject test seemed to show that the adaptive systems provided about the same attenuation of the undesired speaker, but the outputs from the adaptive systems had more reverberation than those from Shields' system.

In the white noise test Shields' system seemed to perform a little better. The "reedy" sound mentioned in Shields' thesis[35] was present in the output. In the adaptive systems the output had a "buzzing" sound that seemed to be more annoying than the distortion in the output from the comb filter. Again the spectrograms indicate no major differences for this test.

The combined signal test has been presented with spectrograms, and

the combined signal spectrogram is shown in figure 7-1 (a). The two waveforms were taken from two different male speakers with approximately the same average pitch. The two signals were aligned in time so that both speakers would be talking at the same time. This alignment would show how well the filtering performed when both speakers were talking. Figures 7-1 (b) and (c) show the spectrograms of the two waveforms before they were added, and the spectrogram in figure 7-1 (b) has been designated as the desired speaker. Figure 7-1 (d) is the spectrogram from the output waveform of Shields' system with seven coefficients, and a Hamming Window as the lowpass prototype. Finally, figure 7-1 (e) is a spectrogram of the adaptive system's output waveform. The output waveform in this figure was produced with seven coefficients and a Hamming Window as the lowpass prototype also. Both figures 7-1 (d) and 7-1 (e) are very similar with each one having only minor differences from the other. This type of result was normally encountered in other sets of sentences that were examined.

As mentioned earlier in this section, the adaptive systems produced a characteristic buzz. The origin of this problem was investigated by examining the input and output waveforms in the areas where the buzzing sound was most prominent. These waveforms are shown in figure 7-2. In the output waveform it can be observed that the major peak in the input waveform is being clipped and distorted by the adaptive system. This phenomenon was not observed in Chapter V when working with test signals. In the case of the test signals, the pitch period was allowed to change very rapidly in some areas, and the adaptive filter performed satisfactorily. It should be pointed out

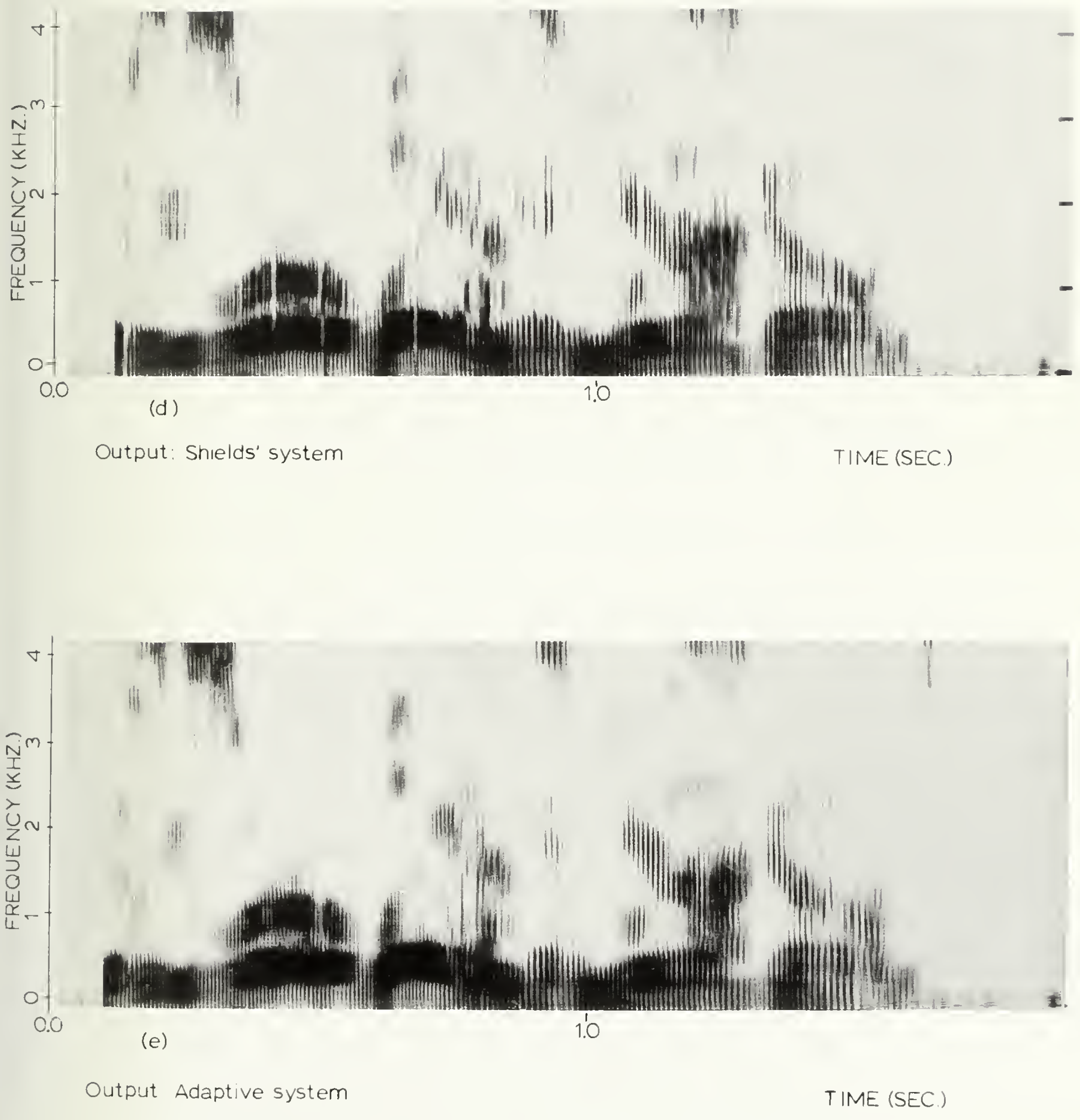


Combined signal

"We were away a year ago"; desired speaker

"There's a chance it will happen once more"; noise speaker

FIGURE 7-1 COMBINED SIGNAL TEST



(d)

Output: Shields' system

(e)

Output Adaptive system

FIGURE 7-1 COMBINED SIGNAL TEST (CONTINUED)

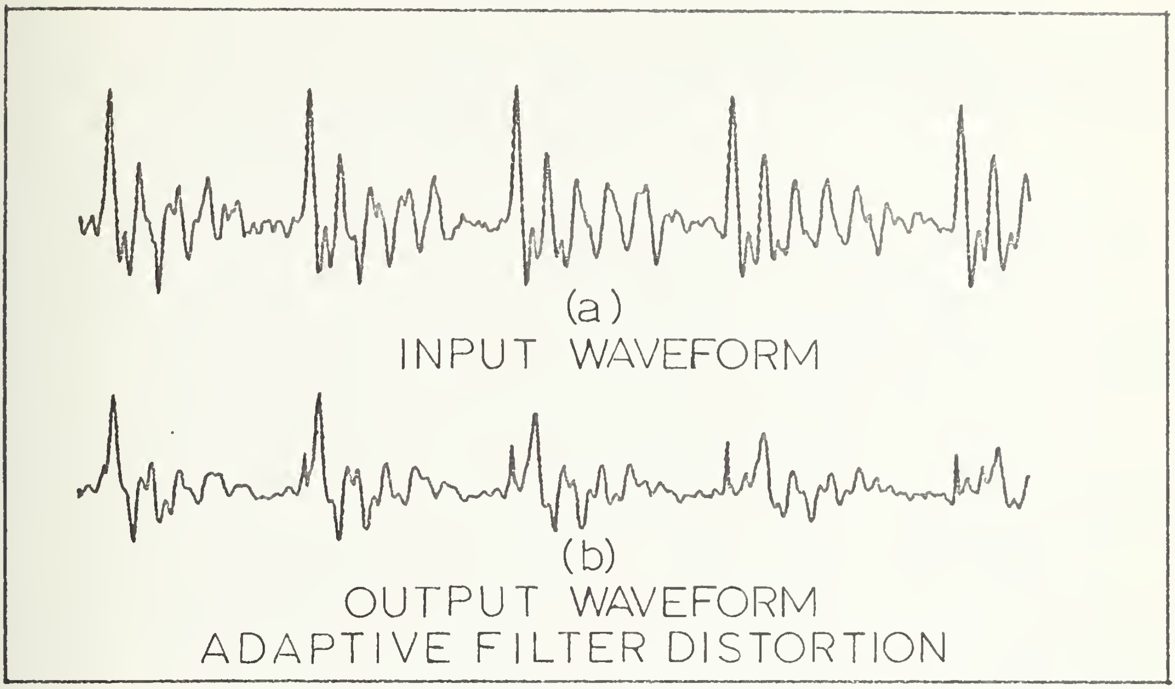


Figure 7-2

that in the case of the test signals, the impulse response of the vocal tract did not change. Therefore, the distortion that is shown in figure 7-2 (b) may be due to the increased variation of the impulse response in short intervals of time. This phenomenon, fortunately, does not occur frequently in normal speech, but it may occur often enough to be an annoying problem.

7.2 Notch Filter Results with Speech Waveforms

The notch filters were implemented to reject the unwanted speaker. Several sentences were processed, and the overall opinion of the notch filter was that it did not improve the quality of the output waveforms. The notch filter was implemented by itself and in series with a comb filter. The notch filter or series combination, at best, performed

only as well as the single comb filter by itself. In most cases the results of the notch filter or series combination of notch and comb were worse than those of the single comb.

The distortion of the desired speaker was higher in the notch filter implementation. This increase in distortion was probably due to the wide transistion width that resulted from the design specifications. The series combination also had the problem of "harmonic overlap". The "harmonic overlap" problem may be explained in the following manner: If the teeth of the comb filter are set to pass the desired speaker and the notches of the notch filter are set to attenuate the unwanted speaker, there may be some areas in the frequency domain when the teeth and notches will overlap. This problem can occur even when the pitch of the two voices are very different. The overall result of this combination of notch and comb has to be lower in performance when the "harmonic overlap" occurs. Because of the results from several processed sentences the notch filter implementation was not pursued further.

In this chapter the descriptions of the speech enhancement systems that were implemented were discussed. This chapter was included for the purposes of showing that the systems did in fact provide some type of speech enhancement. These systems will be tested in another effort, and at that time, some more analytical results may be revealed. A demonstration tape for all of the tests described in this section along with other sentences was made and may be found in the Digital Signal Processing Group Library, M. I. T..

CHAPTER VIII

RESULTS AND OBSERVATIONS

The next two chapters will summarize the results and conclusions of this thesis. Chapter VIII will deal with the overall results while Chapter IX will cover the opinions and general conclusions.

The general goals of this thesis were fulfilled. The system that was described by Shields has been fully implemented on the computer and was tested thoroughly for correct behavior. As an alternative to the comb filtering methods, an adaptive system, has been proposed, implemented, and tested. A limited amount of comparison was performed with test signals and actual speech waveforms.

The adaptive methods can be summarized in the following manner: The method originated from the time-varying structure of voice pitch. The filter was modified to conform to the variations in pitch in order to pass the desired waveform with as little distortion as possible. It was shown in Chapter V that the adaptive system was able to pass a time-varying speech-like waveform when the impulse response of the vocal tract was constant. From this viewpoint, it was shown that the adaptive methods surpassed the methods formulated by Shields. On the other hand, it was stated in Chapter VII that the adaptive systems had some problems when the impulse response varied rapidly in short intervals of time. It should also be pointed out that the adaptive system

was more complex and more time-consuming in the computer implementation than the system proposed by Shields.

The adaptive system stressed fidelity of the desired speaker, and apparently does no worse on the rejection of the unwanted waveform.

Finally, it was observed that the notch filter and the notch and comb filters operating in series did not provide any substantial improvements in the overall quality of the output waveform. In fact, both of these methods showed substantial degradation in some cases.

CHAPTER IX

CONCLUSIONS

In concluding this discussion the following items should be stressed: First, the method proposed in this thesis, the adaptive filter, works well in some cases in the speech enhancement problem. In other cases, the method has its limitations. The good and bad points of the adaptive system were summarized in Chapter VIII. There is little doubt that the adaptive filter reduces the level of the undesired speaker in all cases. The desired waveform was sometimes distorted and occasionally unintelligible.

Second, the method used for pitch detection as developed in Chapter III appears to be a very good approach for the pitch detection problem. The general pitch detection problem was not considered in this thesis, but the entire field of automatic pitch detection algorithms has been the topic of much research for many years. This method may possibly be improved by first taking the derivative of the glottal waveform, and then, using the peak picking algorithm on the resulting waveform. There is still one major drawback in this method: The transition areas between voiced and unvoiced speech are not easily marked whether automatic or manual processing is used. There is a general lack of structure in these areas, and a decision has to be made as to whether or not a mark is needed and where the mark should

be placed. This method of pitch detection also proved to be proficient in the automatic detection of silent areas. Although the maximum benefit of knowing where these were located was not taken advantage of, the silent area detection helped speed up the processing at the beginning and ending of the sentences. Overall, this method was generally accurate and swift. The feasibility of the method may be in question due to the fact that the glottal waveform is not as accessible as the speech waveform.

Third, it can be concluded that the notch filter used to reject the unwanted speaker does not provide substantial overall improvement when both attenuation of undesired signals and faithful reproduction of the desired signals are considered. This conclusion is not surprising, because the notch filter is basically the dual of the comb filter. Based on these observations, the notch filter should not be used because it introduces the serious distortion of the desired waveform. The conclusion is that it seems more feasible to pass the desired speaker with as little distortion as possible than to attenuate the unwanted speaker to a higher degree.

A major observation that should be examined is the question of how effective these methods are in speech enhancement. It was concluded that for normally mixed speech waveforms, there is a limit to the amount of enhancement that can be achieved with either of the filtering methods discussed. It is believed that even optimally designed filters for specific speakers or even for specific sentences, would only provide limited improvement because of inherent overlap

in the signal spectra.

The remaining areas open for improvement in this type of approach to the problem seem to be in the area of polishing the adaptive system so that it approaches this limit. The difficulty in this area lies with the fact that much more information concerning the speech waveform will have to be revealed before the techniques can be improved. Until that time, alternative methods should be examined for the problem of speech enhancement.

FOOTNOTES

[1] Shields, Vaden C. Jr., "Separation of Added Speech Signals by Digital Comb Filtering", M.I.T. Master's Thesis, M.I.T., September, 1970.

[2] Flanagan, James L., Speech Analysis, Synthesis, and Perception, Springer-Verlag, New York, 2nd edition, 1972, p.11.

[3] Oppenheim, A. V. and Schafer, R. W., Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1975, p.512.

[4] Fant, G., Acoustic Theory of Speech Production, s'Gravenhage, Mouton and Company, 1960, p.17.

[5] Shaffer, H. L., "Information Rate Necessary to Trasmit Pitch Period Duration of Connected Speech", Journal of Acoustical Society of America , Volume 36, October, 1964, pp. 1895-1900.

[6] Oppenheim and Schafer, op. cit., p.512.

[7] Ibid., (Figure used with permission of author).

[8] Ibid. p.513.

[9] Shields, op. cit., p. 5.

[10] Ibid., p. 9.

[11] Ibid.

[12] Ibid., pp. 9-10.

[13] Ibid., p. 10.

[14] Ibid., p.12.

[15] Ibid., pp. 13-15.

[16] Ibid., pp. 15-16.

[17] Ibid., p. 16.

[18] Ibid., p. 17.

[19] Ibid., pp. 18-19.

[20] Ibid., p. 20.

[21] Ibid., pp. 20-21.

[22] Ibid., p. 21.

[23] Ibid., p.22.

[24] Ibid., pp.25-26.

[25] Parsons, Thomas, W., "Automatic Separation of Simultaneous Speech of Two Talkers", Paper Z-16, 88th Meeting of Acoustical Society of America, November 8, 1974, pp. 1-6.

[26] Schroeder, M. R., "Parameter Estimation in Speech: A Lesson in Unorthodoxy", Proceedings of the Institute of Electrical and Electronics Engineers, Volume 58, No. 5, May, 1970, pp. 707-712.

[27] Parsons, op. cit., pp. 2-3.

[28] Ibid., p. 5.

[29] Ibid., p. 4.

[30] Henke, William L., "Signals From External Accelerometers During Phonation: Attributes and Their Internal Correlates", M.I.T. Research Laboratory of Electronics, Quarterly Progress Report, July, 15, 1974, pp. 224-231.

[31] Ibid., p. 229.

[32] Shields, op. cit., p. 21.

[33] Ibid., p. 27.

[34] Ibid., p. 16.

[35] Ibid., p. 21.

BIBLIOGRAPHY

Fant, G. Acoustic Theory of Speech Production. 's-Gravenhage, Mouton and Company. 1960.

Flanagan, James L. Speech Analysis, Synthesis and Perception. Springer-Verlag, New York. (2nd edition), 1972.

Henke, William L. "Signals From External Accelerometers During Phonation: Attributes and Their Internal Correlates". M.I.T. Research Laboratory of Electronics, Quarterly Progress Report, pp. 224-231. July 15, 1974.

Oppenheim, Alan V., and Schafer, Ronald W. Digital Signal Processing. Prentice-Hall Inc., Englewood Cliffs, N.J. 1975.

Parsons, Thomas W. "Automatic Separation of Simultaneous Speech of Two Talkers". Paper Z-16, 88th Meeting of the Acoustical Society of America. November 8, 1974.

Schroeder, M. R. "Parameter Estimation in Speech: A Lesson in Unorthodoxy". Proceedings of the Institute of Electrical and Electronics Engineers. Volume 58, No. 5, pp.707-712. May, 1970.

Shaffer, H. L. "Information Rate Necessary to Transmit Pitch Period Durations of Connected Speech". Journal of the Acoustical Society of America. Volume 36, pp. 1895-1900. October, 1964.

Shields, Vaden C,, Jr. "Separation of Added Speech Signals by Digital Comb Filtering". M.I.T. Master's Thesis. September, 1970.

APPENDIX

A.1 Introduction

This appendix contains the actual computer programs that were used in implementing the systems described in this thesis. They were included as an appendix for several reasons. First, if these systems need to be implemented again, these programs will provide a reasonable starting point. Second, if these systems are to be compared with other systems, there will be no question as to whether or not the systems described in this thesis have been correctly implemented. Finally, these programs were placed in the thesis instead of a separate notebook in order that the two sections would not become separated.

These programs were written in RT-11 Fortran, and all non standard fortran routines will be included for completeness. It should also be mentioned that some of the features of RT-11 Fortran may not translate directly into statements usable with another Fortran compiler, but for the most part, the RT-11 Fortran is compatible with other versions.

The PDP-11 structure stresses the use of modularity in programming, and this concept has been used to a great extent in these implementations. For each module or program, a brief flowchart is included with the computer code. This will probably be a helpful feature if the programs in this appendix are to be dissected.

Figure A-1 shows a general layout of the overall systcms with the modules shown as blocks and the computer file names that are used as inputs and outputs to these blocks. Figure A-1 is used in the processing of waveforms exactly as shown if there are no pitch errors to be corrected. However, if pitch errors occur, then these errors have to be corrected as shown in figure A-2.

These computer programs were not written with speed of execution as a primary concern, but with user interaction capability and comprehension for new users as the primary concerns.

COMPUTER PROGRAM INDEX

COMPUTER PROGRAM INDEX
(Continued)

COMPUTER SYSTEM LAYOUT

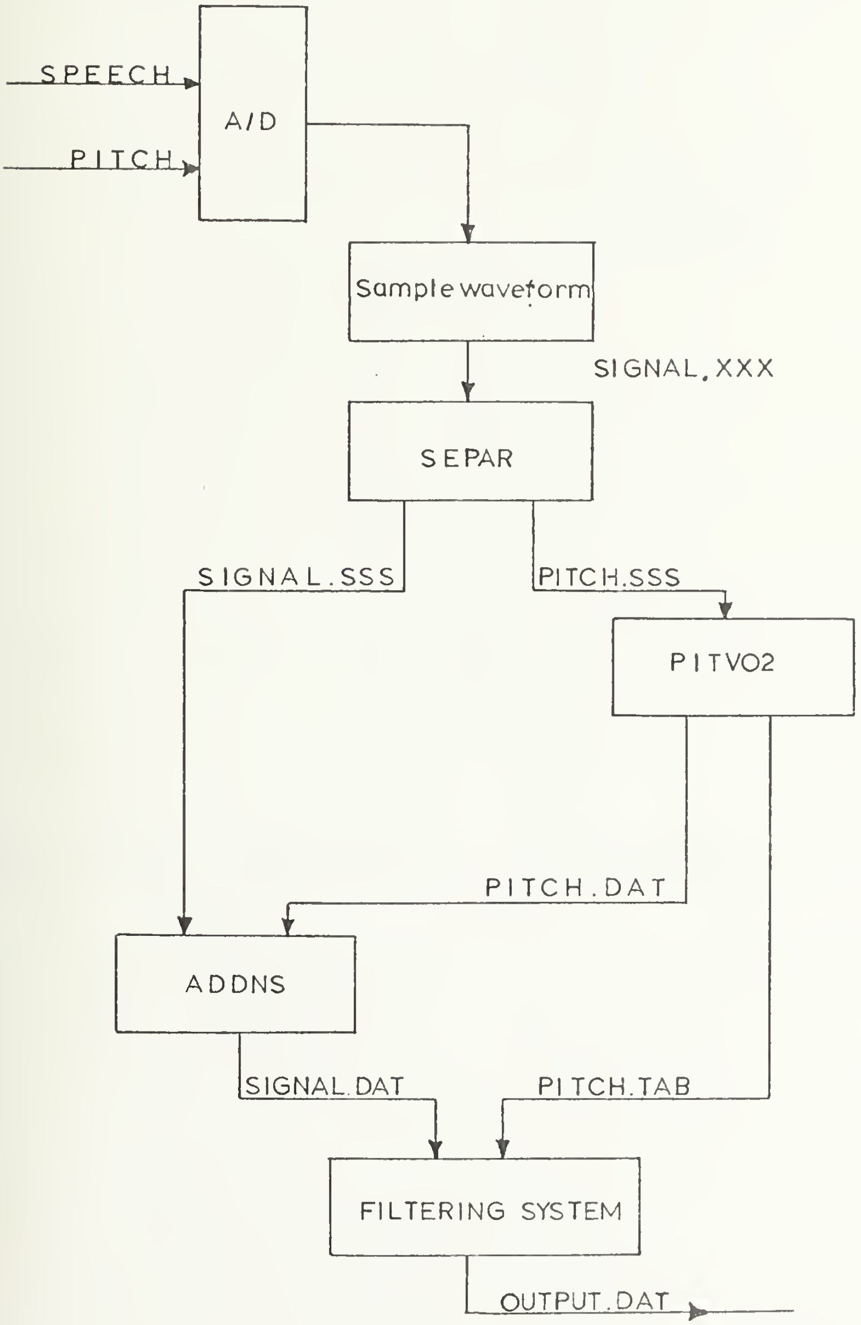


FIGURE A-1

CORRECTION OF PITCH MARKS

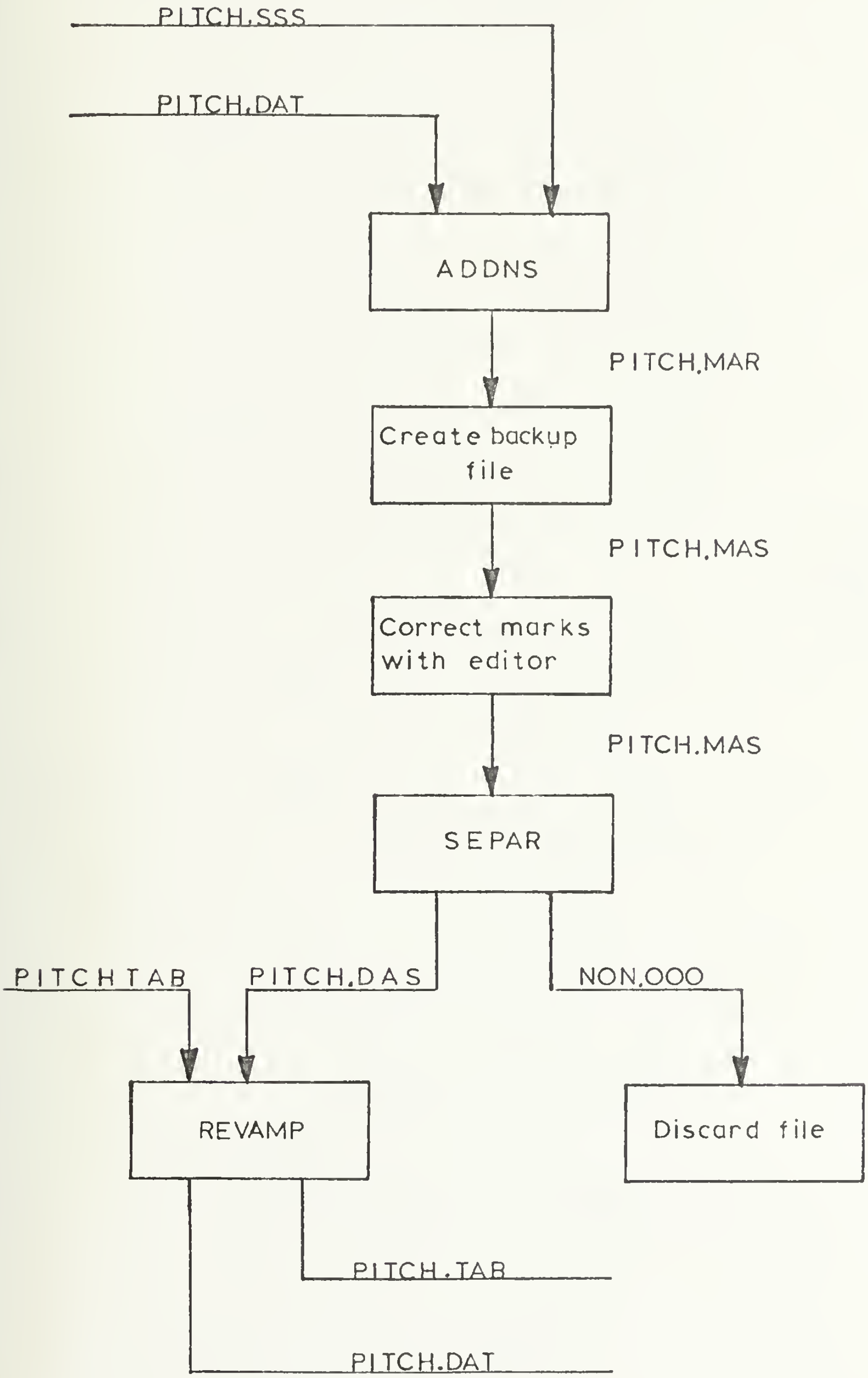


FIGURE A-2

ADDS

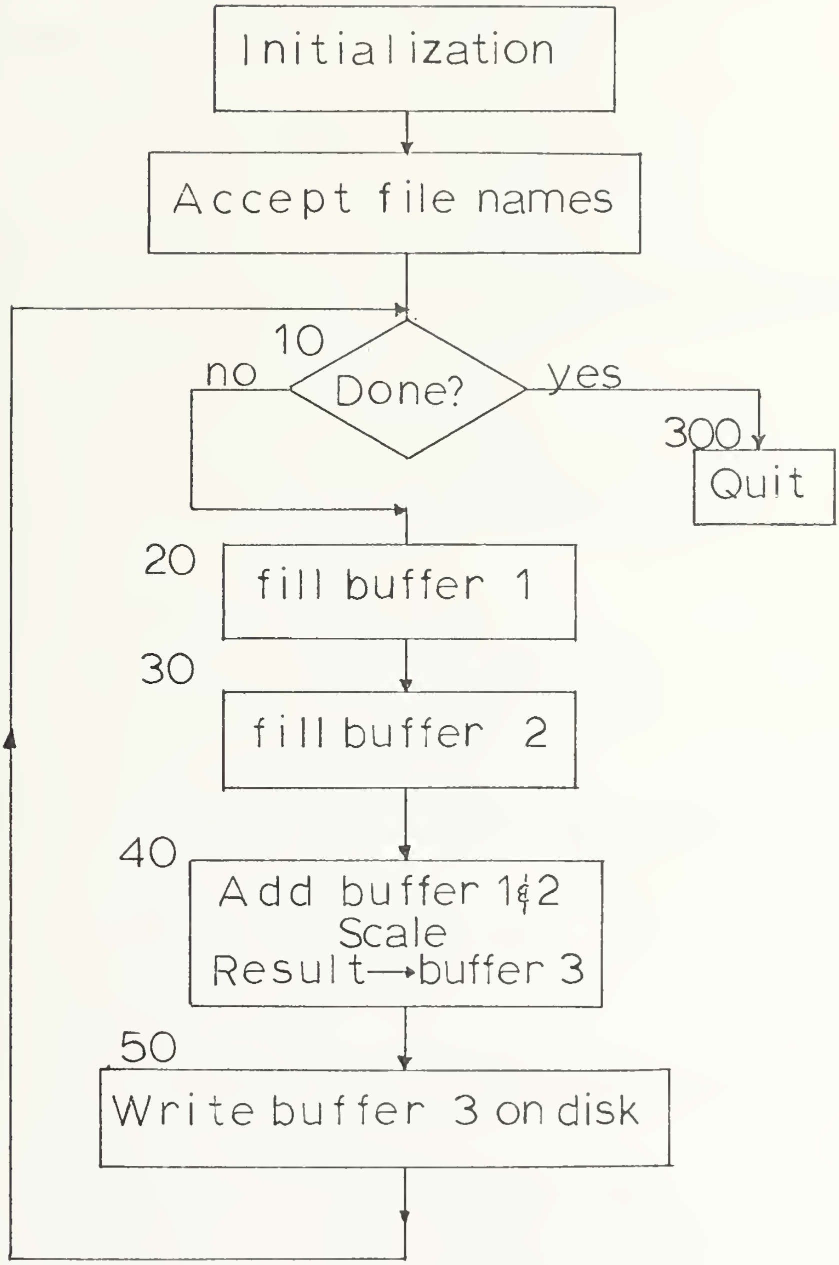


FIGURE A-3

```
C
C
C       TITLE:  ADDS.FOR 750122
C
C THIS PROGRAM ADDS TWO WAVEFORMS TOGETHER ON THE
C ODD CHANNELS, AND DIVIDES THE SUM BY 2 TO
C PREVENT OVERFLOW.
C
C       INITIALIZATION
        INTEGER DATBUF(2560),CATBUF(2560),BATBUF(2560)
        DATA IVAR,JVAR,KVAR/2,2,2/
        WRITE (7,1)
1       FORMAT(' ','NAME OF FILE 1 <FILNAM.EXT>')
        WRITE(7,2)
2       FORMAT(' ',' ')
        CALL ASSIGN (2,'RK1:SIGNAL.XXX',-15)
        WRITE(7,3)
3       FORMAT(' ','NAME OF FILE 2 <FILNAM.EXT>')
        WRITE (7,2)
        CALL ASSIGN (3,'RK1:SIGNAL.SSS',-15)
        WRITE(7,4)
4       FORMAT(' ','NAME OF OUTPUT FILE <FILNAM.EXT>')
        WRITE(7,2)
        CALL ASSIGN (4,'RK1:SIGNAL.DAT',-15)
        DEFINE FILE 2 (251,256,U,IVAR)
        DEFINE FILE 3 (251,256,U,JVAR)
```

```
      DEFINE FILE 4 (251,256,U,KVAR)
C PROCESSING BEGINS HERE.
10    IF(IVAR.EQ.252) GO TO 300
C FILL BUFFERS.
      DO 20 IA=1,10
      READ(2'IVAR) (DATBUF(IB + ((IA-1) * 256)),IB=1,256)
20    CONTINUE
      DO 30 IC=1,10
      READ(3'JVAR) (CATBUF(ID + ((IC-1) * 256)),ID=1,256)
30    CONTINUE
C ADD THE TWO TOGETHER.
      DO 40 IE=1,2559,2
      BATBUF(IE)=(DATBUF(IE) + CATBUF(IE))/2
      IZ=IE+1
      BATBUF(IZ)=0
40    CONTINUE
C WRITE THE OUTPUT
      DO 50 IF = 1,10
      WRITE (4'KVAR) (BATBUF(IG + ((IF-1) * 256)),IG=1,256)
50    CONTINUE
      GO TO 10
300   ENDFILE 2 !TURN FILES OFF
      ENDFILE 3
      ENDFILE 4
      STOP
      END
```

SEPAR

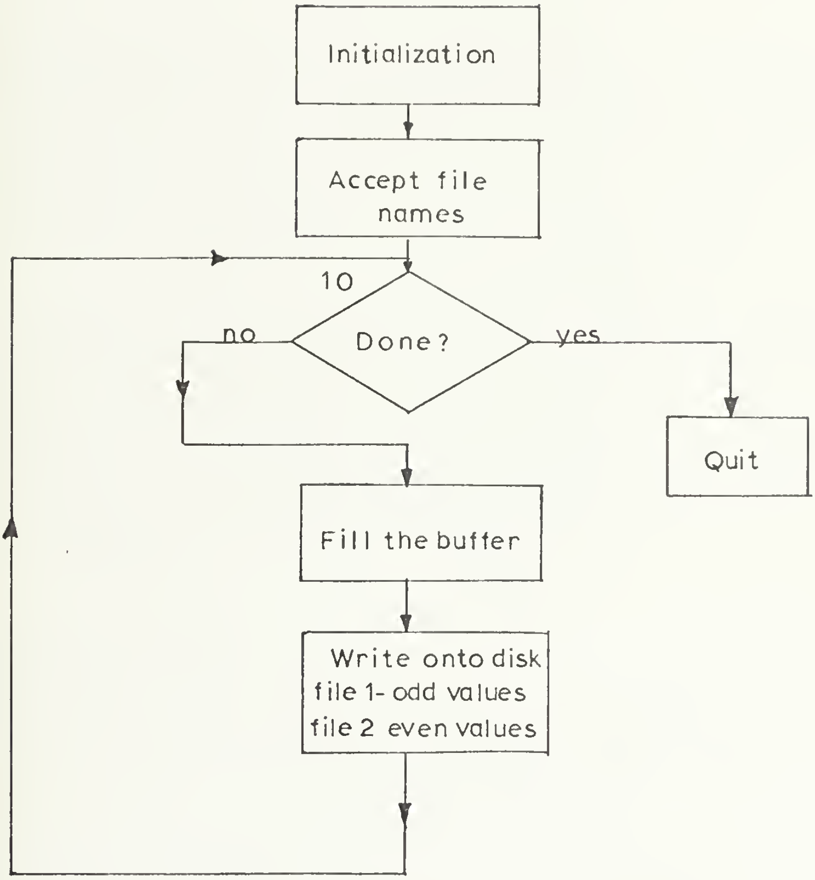


FIGURE A-4

```
C
C
C       TITLE:  SEPAR.FOR       750121
C
C
C THIS PROGRAM IS USED TO SEPARATE THE TWO CHANNELS OF A
C FILE AND WRITE THEM INTO INDIVIDUAL FILES WITH NO DATA
C ON THE EVEN CHANNELS.
C
C IF SWITCH(0) IS UP THE PROGRAM ALLOWS NAMED FILES TO BE
C TYPED IN.  IF SWITCH (1) IS UP THE OUTPUT 2 FILE IS IN
C EVEN FORMAT.  IF SWITCH (1) IS DOWN THE OUTPUT 2 FILE IS
C IN ODD FORMAT.
C
C SUBROUTINES USED
C JSSWI (SEE REVAMP PROGRAM FOR EXPLANATION.)
C
C       INITIALIZATION
        INTEGER DATBUF(2560), CATBUF(2560), BATBUF(2560)
        DATA IVAR,JVAR,KVAR/2,2,2/
        WRITE(7,1)

        WRITE(7,2)
2       FORMAT(' ','SWITCH(0) DOWN - DEFAULT FILES USED')
        WRITE(7,3)
3       FORMAT(' ','INPUT - SIGNAL.XYX')
```

```
      WRITE(7,4)
4     FORMAT(' ','OUTPUTS -SIGNAL.SSS  PITCH.SSS')
      WRITE(7,5)
5     FORMAT(' ','SWITCH(0) UP - I/O FILES MAY BE CHOSEN')
      WRITE(7,6)
6     FORMAT(' ','SWITCH(1) DOWN-OUTPUT 2 IN ODD CONFIG.')
      WRITE(7,7)
7     FORMAT(' ','SWITCH(1) UP - OUTPUT 2 IN EVEN CONFIG.')
      PAUSE 'PAUSE'
      IF (ISSWI(0).EQ.1) GO TO 8
      CALL ASSIGN (2,'RK1:SIGNAL.XXX')
      CALL ASSIGN (3,'RK1:SIGNAL.SSS')
      CALL ASSIGN (4,'RK1:PITCH.SSS')
      GO TO 15
8     WRITE (7,9)
9     FORMAT(' ','NAME OF INPUT FILE <FILNAM.EXT>')
      WRITE(7,10)
10    FORMAT(' ',' ')
      CALL ASSIGN (2,'RK1:SIGNAL.SSS',-14)
      WRITE(7,11)
11    FORMAT(' ','NAME OF OUTPUT FILE 1 <FILNAM.EXT>')
      WRITE (7,10)
      CALL ASSIGN (3,'RK1:SIGNAL.001',-14)
      WRITE (7,12)
12    FORMAT(' ','NAME OF OUTPUT FILE 2 <FILNAM.EXT>')
      WRITE(7,10)
```

```
      CALL ASSIGN (4,'RK1:SIGNAL.002',-14)
15    DEFINE FILE 2 (251,256,U,IVAR)
      DEFINE FILE 3 (251,256,U,JVAR)
      DEFINE FILE 4 (251,256,U,KVAR)
C ARE WE THRU?
17    IF(IVAR.GE.252) GO TO 300
C FILL THE BUFFER.
      DO 20 IA = 1,10
      READ(2'IVAR) (DATBUF(IB + ((IA-1) * 256)),IB=1,256)
20    CONTINUE
C FILL THE ARRAYS - CHANNEL(4) = ODD POINTS;
C                   CHANNEL(14) = EVEN POINTS.
      DO 30 IC=1,2559,2
      DATBUF(IC)=DATBUF(IC)
      DATBUF(IC + 1) = 0
30    CONTINUE
      DO 40 ID=1,2559,2
      IF (ISSWI(1).EQ.0) DATBUF(ID) = DATBUF(ID + 1)
      IF (ISSWI(1).EQ.0) DATBUF(ID + 1) = 0
      IF(ISSWI(1).EQ.1) DATBUF(ID) = 0
      IF (ISSWI(1).EQ.1) DATBUF(ID + 1) = DATBUF(ID+1)
40    CONTINUE
C WRITE THE OUTPUT ONTO THE SPECIFIED FILES.
      DO 50 IE = 1,10
      WRITE(3'JVAR) (DATBUF(IF + ((IE-1) * 256)),IF=1,256)
50    CONTINUE
```

```
      DO 60 IG= 1,10
      WRITE(4'KVAR) (BATLLF(IH + ((IG-1) * 256)),IH=1,256)
60    CONTINUE
      GO TO 17
500   ENDFILE 2 !TURN FILES OFF.
      ENDFILE 3
      ENDFILE 4
      STOP
      END
```

PITVO2

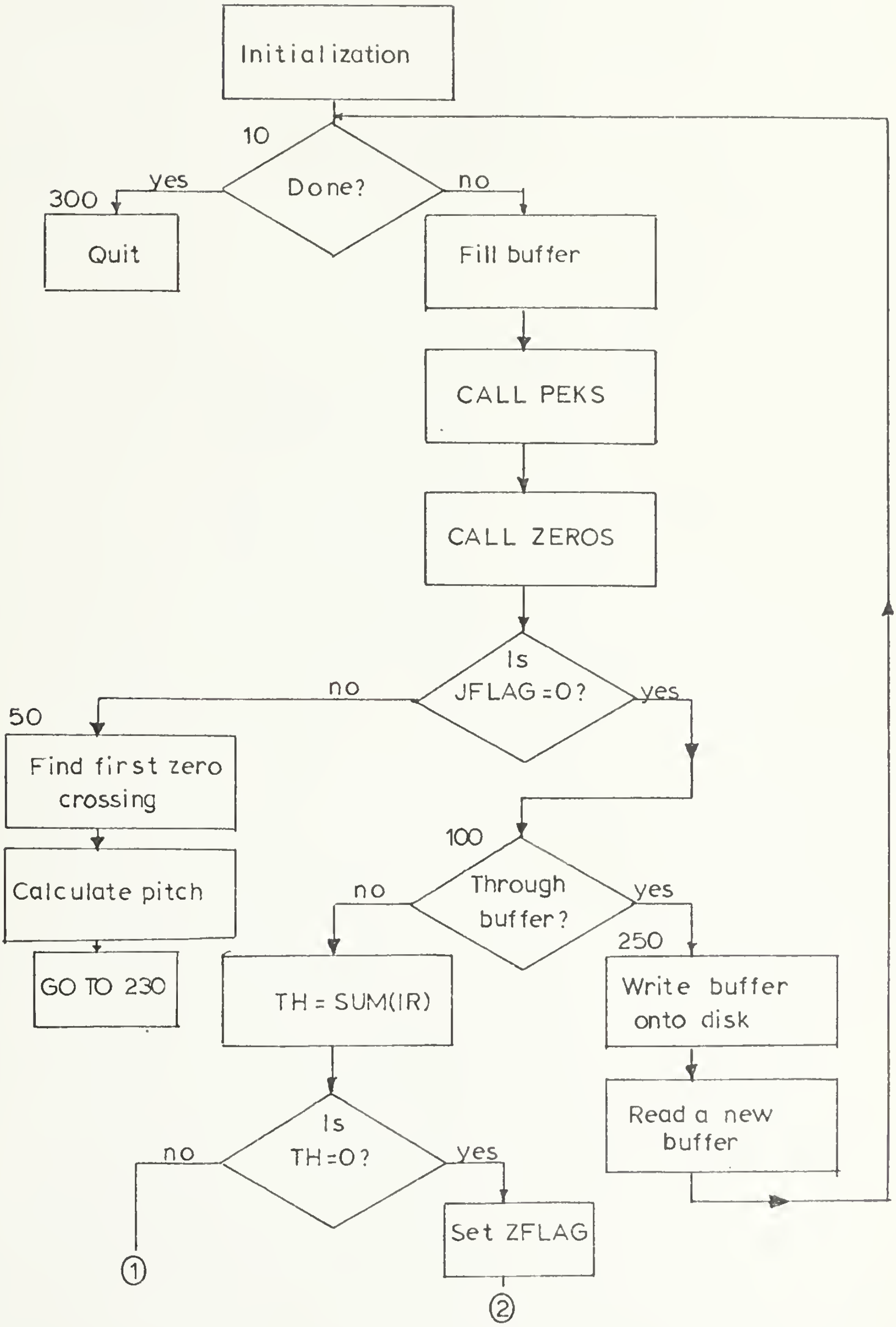


FIGURE A-5

PITVO2 (CONT.)

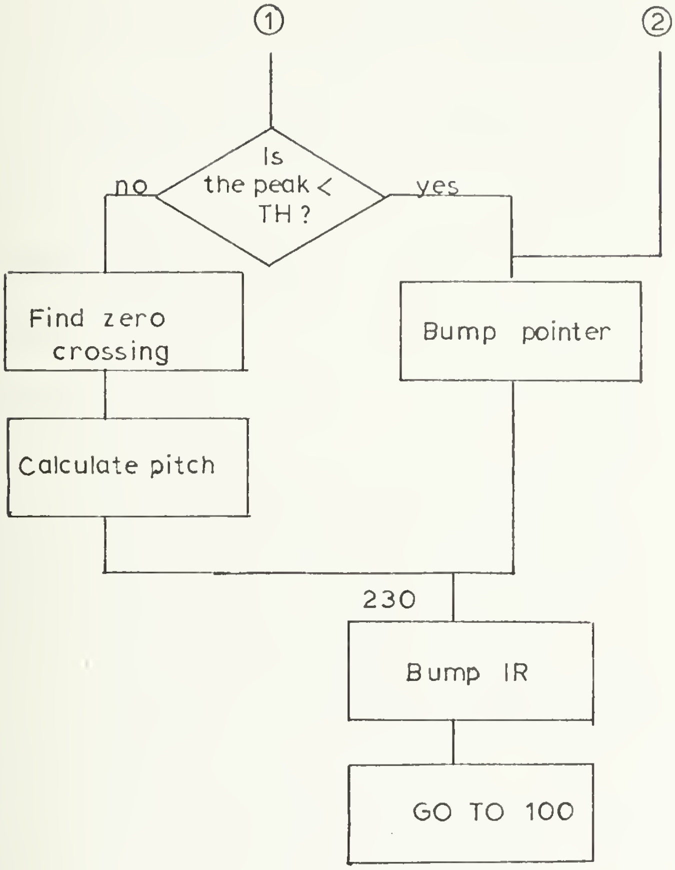


FIGURE A-5(CONT.)

```
C
C
C
C       TITLE:  PITVO2   750220
C
C SUBROUTINES USED:
C PEKS
C ZEROS
C
C PARAMETERS USED AND DEFINITION:
C JFLAG:  THIS IS USED TO DENOTE AN OVERFLOW IN THE BUFFER.
C         IT OCCURS WHEN A PEAK IS DETECTED AT THE END OF A
C         BUFFER, AND THE CORRESPONDING ZERO CROSSING
C         HAS NOT BEEN FOUND.  WHEN JFLAG IS SET (=1)
C         THE PROGRAM LOOKS FOR A ZERO CROSSING IMMEDIATELY
C         AFTER A BUFFER SHIFT HAS BEEN PERFORMED.
C
C ZFLAG:  THIS FLAG IS USED TO DENOTE A SILENT AREA
C         AND WHEN SET (=1) A PITCH PERIOD IN THE PITCH
C         TABLE IS GIVEN A NEGATIVE SIGN TO DENOTE SILENCE.
C
C INITIALIZATION
C
      INTEGER DATBUF(2560),A(1280),B(1280),BUF(2560)
      INTEGER SUM(5),ZFLAG, START, TH, END1
      COMMON DATBUF
```

```
      DATA BUF,I,J,START/2560*0,1,1,1/
      DATA IVAR,JVAR,KVAR,IR,ZFLAG,JFLAG/2,2,2,1,0,0/
      CALL ASSIGN (2,'RK1:PITCH.SSS')
      CALL ASSIGN (3,'RK1:PITCH.TAB')
      CALL ASSIGN (4,'RK1:PITCH.DAT')
      DEFINE FILE 2 (251,256,U,IVAR)
      DEFINE FILE 3 (1000,2,U,JVAR)
      DEFINE FILE 4 (251,256,U,KVAR)
10    IF (IVAR.EQ.252) GO TO 300      !DO WE QUIT?
      I=1         !NO, RESET THE POINTER.
C FILL THE BUFFER
      DO 20 II=1,10
      READ(2'IVAR) (DATBUF(J + ((II-1) * 256)),J=1,256)
20    CONTINUE
C FIND THE PEAKS AND ZERO CROSSINGS.
      CALL PEKS (A,B,IVAR,SUM)
      CALL ZEROS (A,B)
C CALCULATE PITCH
C IS THE OVERFLOW FLAG SET?
      IF (JFLAG.EQ.0) GO TO 100
C THIS SECTION PICKS UP THE ZERO CROSSING FOR AN OVERFLOW.
      J=1         !YES, FIND THE FIRST ZERO CROSSING.
50    IF(A(J).NE.0) GO TO 70
      J = J + 1
      IF(J.GT.1280) GO TO 250
      GO TO 50
```

```
70      END1=J              !FOUND IT.
        M=END1-START        !CALCULATE THE PITCH PERIOD.
        START=END1          !RESET THE START POINTER.
        INDEX=2*J !CALCULATE POITION.
        BUF(INDEX) = 4095         !MARK THE PITCH EPOCH.
        KU = INDEX/256      !THIS SECTION DETERMINES THE
        IF((KU*256).EQ.INDEX) KU = KU - 1 !REC. NO. THAT
        IREC = KU + KVAR !CORRESPONDS TO THE SPEECH FILE.
        IF (IREC.GT.251) IREC = 251
        IF (ZFLAG.EQ.0) GO TO 80 !IS THIS A SILENT AREA?
        M = -1 * M !YES, CHANGE PITCH PERIOD TO A NEG. VALUE.
        ZFLAG = 0 !CLEAR THE SILENT FLAG.
80      WRITE(3'JVAR) M, IREC
        I=J
        LFLAG=0             !CLEAR THE OVERFLOW FLAG.
        GO TO 230
100     IF(I.GT.1280) GO TO 250 !NO, ARE WE THRU?
        TH = SUM (IR)       !OBTAIN THRESHOLD.
        IF (TH.NE.0) GO TO 120
        ZFLAG = 1
        GO TO 200
120     IF(B(I).LT.TH) GO TO 200 !NO, IS THE PEAK VALUE< TH?
        J=I                 !NO, GO CHECK ZERO CROSSINGS.
150     IF (A(J).NE.0) GO TO 180      !ZERO CROSSING?
        J=J+1               !NO, BUMP THE COUNTER.
        IF (J.LE.1280) GO TO 150      !THRU BUFFER?
```

```
        START=START-1280  !YES,STORE START LOCATION.
        JFLAG=1           !SET THE OVERFLOW FLAG.
        GO TO 250         !GO READ THE BUFFER.
180     END1 =J           !YES, ZERO CROSSING OCCURRED.
        M=END1-START      !CALCULATE THE PITCH.
        INDEX=2 * J       !CALCULATE POSITION FOR MARK.
        BUF(INDEX) = 4095 !MARK THE PITCH EPOCH.
        KU = INDEX/256    !CALCULATE RECORD NUMBER
        IF ((KU*256).EQ.INDEX) KU = KU - 1
        IREC = KU + KVAR  !OF SPEECH WAVEFORM.
        IF (IREC.GT.251) IREC = 251
        IF (ZFLAG.EQ.0) GO TO 190 !IS THIS A SILENT AREA?
        M = -1 * M        !YES, DENOTE WITH NEG. VALUE.
        ZFLAG = 0
190     WRITE(3'JVAR) M, IREC
        START = END1      !RESET THE START POINTER.
        I = I + 20        !RESET I BY MINIMUM PITCH.
        GO TO 230
200     I=I+1
230     IR = I/256
        IF ((IR * 256).NE. I) IR = IR + 1
        GO TO 100
250     IF (JFLAG.EQ.0) START=START-1280
C WRITE OUT THE BUFFER ONTO THE FILE ON THE DISK.
        DO 260 IA=1,10
        WRITE(4'KVAR) (BUF(ID + ((IA -1)*256)),ID=1,256)
```

```
260     CONTINUE
C CLEAR THE BUFFER, BUF, FOR THE NEXT CALL.
        DO 270 IC=1,2560
        BUF(IC) = 0
270     CONTINUE
        GO TO 10
300     ENDFILE 2
        ENDFILE 3
        ENDFILE 4
        STOP
        END
```

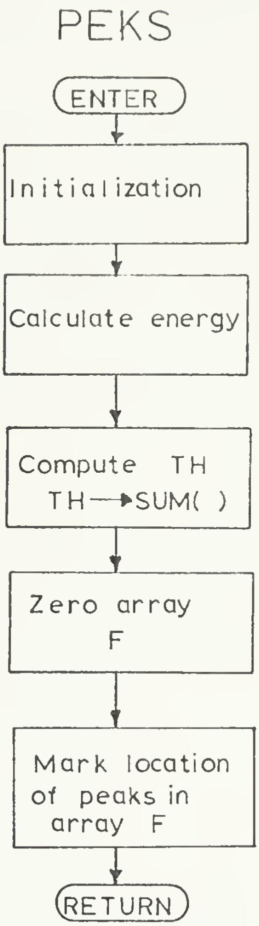

PEKS
ENTER
Initialization
Calculate energy
Compute TH
TH → SUM( )
Zero array
F
Mark location
of peaks in
array F
RETURN

FIGURE A-6

```
C
C
C      TITLE:  PEKS.FOR 750110
C
C THIS SUBROUTINE FINDS THE PEAKS OF A WAVEFORM THAT HAVE
C BEEN STORED IN A BUFFER (DATBUF).  THE RETURN CONTAINS AN
C ARRAY "F" WITH THE VALUE OF THE PEAK IF ONE OCCURRED AND
C WAS GREATER THAN ZERO.  IF NO PEAK OCCURRED AT THAT
C POINT, THEN A ZERO IS RETURNED.
C
C PROGRAM PARAMETERS:
C THE ENERGY SUMMATION IN DO-LOOP 3 HAS A SCALE FACTOR THAT
C IS USED TO PREVENT OVERFLOW.  (IN THIS CASE 100. IS USED
C FOR THE SCALE FACTOR.)
C
C THE ARRAY "SUM" CONTAINS THE THRESHOLDS THAT
C WERE CALCULATED FROM THE ENERGY MEASUREMENT.  THESE
C THRESHOLDS ARE RETURNED TO THE MAIN PROGRAM FOR USE IN
C DETERMINING THE CORRECT PEAKS.
C
C THE THRESHOLD SETTING THAT WAS LINKED TO A PARTICULAR
C MEASUREMENT OF ENERGY WAS DETERMINED BY EXAMINATION OF
C SEVERAL WAVEFORMS AND PLOTTING ENERGY VERSUS  TRUE
C DIGITAL PEAKS.  THIS AREA OF THE PROGRAM MIGHT BE IM-
C PROVED BY A MORE DETAILED STUDY OF THE DATA FOR THE
C OPTIMAL THRESHOLD VALUES.
```

```
C
C PARAMETER "IFLAG" IS USED TO DENOTE A NEGATIVE
C SLOPE.   IFLAG = 1 - NEGATIVE SLOPE
C          IFLAG = 0 - POSITIVE SLOPE
C
      SUBROUTINE PEKS (E,F,IVAR,SUM)
C     INITIALIZATION
      INTEGER FATBUF(2560),E(1280),F(1280),SUM(5)
      COMMON FATBUF
      DO 60 IA = 1,5
C PERFORM AN ENERGY SUMMATION OVER 256 POINTS OF THE
C GLOTTAL WAVEFORM.
      SUMX = 0.0
      DO 3 IB = 1,511,2
      SUMX=SUMX+(((FATBUF(((IA-1)*512)+IB)-2048)/100)**2)
3     CONTINUE
C THIS PART OF PROG. ASSIGNS A TRESHOLD VALUE BASED ON
C THE ENERGY SUMMATION COMPUTED ABOVE.
      IF(SUMX.NE.0) GO TO 10 !SUMX=0 MEANS SILENT AREA
      THR = 0
      GO TO 55
10    IF(SUMX - 12000) 15,40,40
15    IF(SUMX - 6500) 20,35,35
20    IF(SUMX - 1000) 25,30,30
25    THR = 2300
      GO TO 55
```

```
30      THR = 2600
        GO TO 55
35      THR = 3000
        GO TO 55
40      IF (SUMX - 16000) 45,50,50
45      THR = 3400
        GO TO 55
50      THR = 3800
55      SUM (IA) = THR
60      CONTINUE
C FIND THE PEAKS FOR THE SEGMENT OF DATA.
        DO 100 KIT=1,1280
        P(KIT)=0
100     CONTINUE
        K=1
        K2 = 1
130     IF(FATBUF(K+2) - FATBUF(K)) 140,150,160
140     IFLAG = 1
        GO TO 170
150     K = K + 2
        GO TO 130
160     IFLAG = 0
170     DO 210 K=3,2559,2
        P=K-2
        IF (FATBUF(K).GT.FATBUF(P)) GO TO 200
        IF (IFLAG.EQ.1) GO TO 205
```

```
      IF (FATBUF(K).LT.2048) GO TO 205
      F(K2) = FATBUF(K-2)      !FOUND THE PEAK
      IFLAG = 1
      GO TO 205
200   IFLAG=0
205   K2 = K2 + 1
210   CONTINUE
      RETURN
      END
```

ZEROS

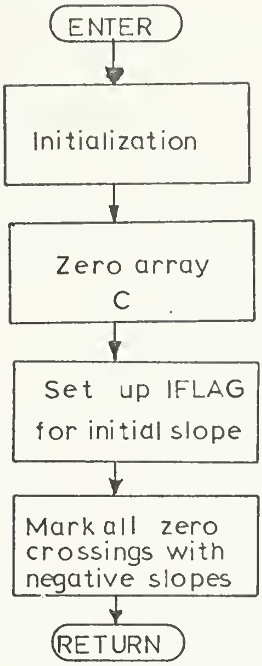

ENTER
Initialization
Zero array C
Set up IFLAG for initial slope
Mark all zero crossings with negative slopes
RETURN

FIGURE A-7

```
C
C
C
C       TITLE:  ZEROS.FOR                 750110
C
C THIS SUBROUTINE FINDS ALL ZERO CROSSINGS OF A WAVE-
C FORM STORED IN A BUFFER.  IF A ZERO CROSSING OCCURS AT
C SOME POINT, THAT POINT IS MARKED WITH A ONE IN THE ARRAY
C "C".  IF NOT, A ZERO IS LEFT IN THE ARRAY.
C
C PARAMETERS USED:
C "IFLAG" IS USED IN THIS SUBROUTINE TO DENOTE AN
C A VALUE OF THE DATA THAT IS NEGATIVE.
C I.E.      IFLAG = 1  DATA VALUE IS NEGATIVE.
C           IFLAG = 0  DATA VALUE IS POSITIVE.
C
        SUBROUTINE ZEROS (C,D)
C
C       INITIALIZATION
        INTEGER CATBUF(2560),C(1280),D(1280)
        COMMON CATBUF
        DO 10 III=1,1280
        C(III) = 0
10      CONTINUE
        K=1
        K2 = 1
```

```
25      IF(CATBUF(N) - 2048) 30,40,50
30      IFLAG=0
        GO TO 60
C MARK A ZERO CROSSING.
40      C(N2) = 1
        N = N + 2
        N2 = N2 + 1
        IF (N.GE.2560) GO TO 200
        GO TO 25
50      IFLAG = 1
60      IF (CATBUF(N) - 2048) 70,40,80
70      IF (IFLAG.EQ.0) GO TO 75
        C(N2) = 1
        IFLAG = 0
75      N = N + 2
        N2 = N2 + 1
        IF (N.GE.2560) GO TO 200
        GO TO 60
80      IF (IFLAG.EQ.1) GO TO 75
        C(N2) = 1
        IFLAG = 1
        GO TO 75
200     RETURN
        END
```

ADDNS

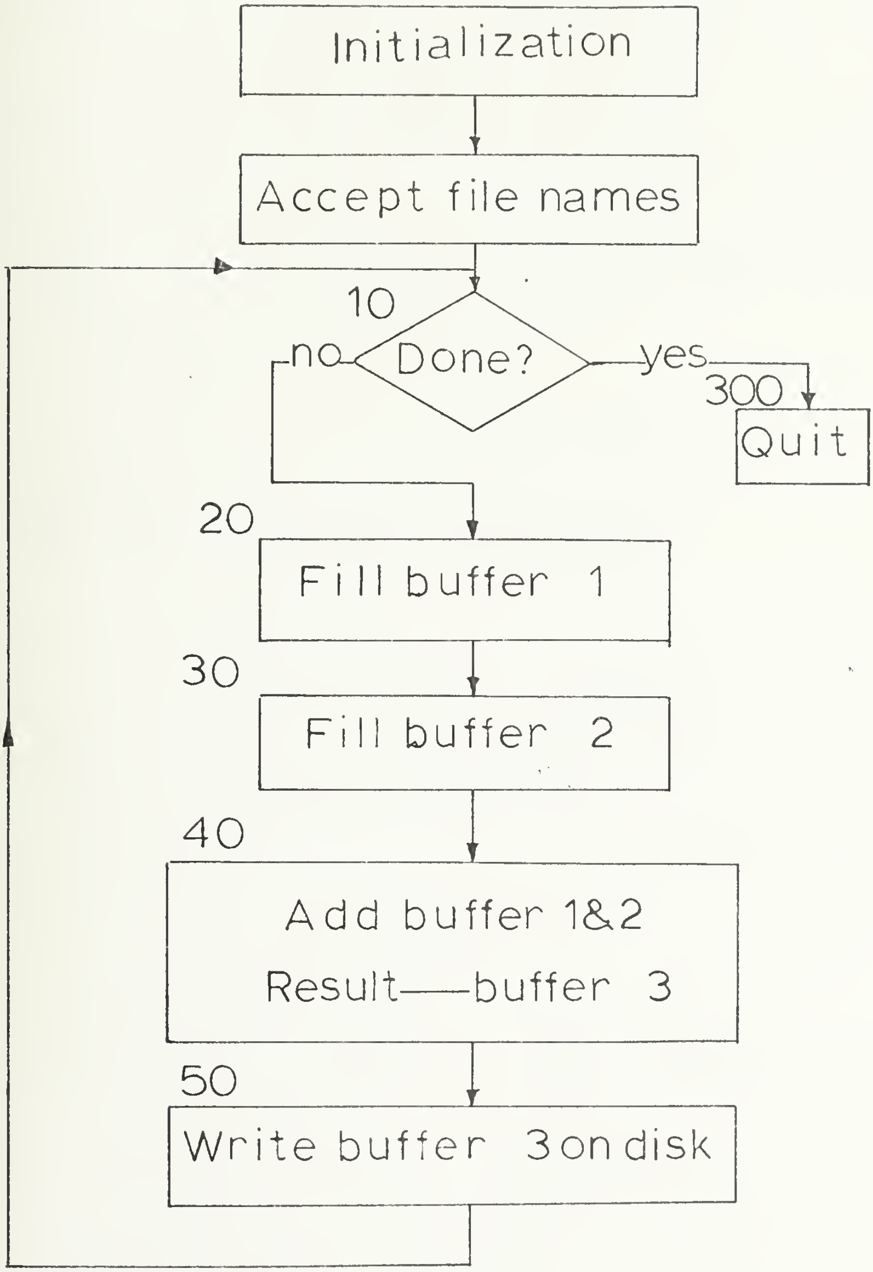


FIGURE A-8

```
C
C
C       TITLE:  ADDNS.FOR       750120
C
C THIS PROGRAM ADDS TWO FILES TOGETHER WITHOUT A SCALE
C FACTOR.  IT IS MEANT TO BE USED ON AN ODD - EVEN FORMAT.
C (I.E. ONE FILE ODD FORMAT, THE OTHER EVEN).  IF SWITCH(0)
C IS UP THE PROGRAM ALLOWS THE INPUT AND OUTPUT FILES TO
C BE NAMES.  IF SWITCH (0) IS DOWN, THEN THE PROGRAM
C  ASSUMES THE DEFAULT NAMES THAT ARE USED IN SERIES WITH
C THE COMPLETE SHIELDS SYSTEM.
C
C SUBROUTINES USED
C ISSWI (SEE REVAMP FOR EXPLANANTION.)
C
C       INITIALIZATION
        INTEGER DATBUF(2560),CATBUF(2560),BATBUF(2560)
        DATA IVAR,JVAR,KVAR/2,2,2/
        WRITE(7,1)
1       FORMAT(' ','PROGRAM CONTROLS - - - SWITCH (0)')
        WRITE (7,2)
2       FORMAT(' ','SWITCH() DOWN - DEFAULT FILES USED')
        WRITE (7,3)
3       FORMAT(' ','INPUT - SIGNAL.SSS')
        WRITE(7,4)
4       FORMAT(' ','OUTPUT - SIGNAL.DAT')
```

```
      WRITE(7,5)
5     FORMAT(' ','SWITCH(0) UP - I/O FILES MAY BE CHOSEN')
      PAUSE 'PAUSE'
      IF (ISSWI(0).EQ.1) GO TO 6
      CALL ASSIGN (2,'RK1:SIGNAL.SSS')
      CALL ASSIGN (3,'RK1:PITCH.DAT')
      CALL ASSIGN (4,'RK1:SIGNAL.DAT')
      GO TO 10
6     WRITE(7,7)
7     FORMAT(' ','NAME OF INPUT FILE <FILNAM.EXT>')
      WRITE(7,8)
8     FORMAT(' ',' ')
      CALL ASSIGN (2,'RK1:SIGNAL.SSS',-14)
      CALL ASSIGN (3,'RK1:PITCH.DAT')
      WRITE (7,9)
9     FORMAT(' ','NAME OF OUTPUT FILE <FILNAM.EXT>')
      WRITE(7,8)
      CALL ASSIGN (4,'RK1:SIGNAL.DAT',-14)
10    DEFINE FILE 2 (251,256,U,IVAR)
      DEFINE FILE 3 (251,256,U,JVAR)
      DEFINE FILE 4 (251,256,U,KVAR)
C PROCESSING BEGINS HERE.
11    IF(IVAR.EQ.252) GO TO 300
C FILL BUFFERS.
      DO 20 IA=1,10
      READ(2'IVAR) (DATBUF(IB + ((IA-1) * 256)),IB=1,256)
```

```
20      CONTINUE
        DO 30 IC=1,10
        READ(3'JVAR) (CATBUF(ID + ((IC-1) * 256)),ID=1,256)
30      CONTINUE
C ADD THE TWO TOGETHER.
        DO 40 IE=1,2559,2
        BATBUF(IE)=DATBUF(IE) + CATBUF(IE)
        BATBUF(IE+1) = CATBUF(IE+1)
40      CONTINUE
C WRITE THE OUTPUT
        DO 50 IF = 1,10
        WRITE (4'KVAR) (BATBUF(IG + ((IF-1) * 256)),IG=1,256)
50      CONTINUE
        GO TO 11
300     ENDFILE 2 !TURN THINGS OFF.
        ENDFILE 3
        ENDFILE 4
        STOP
        END
```

REVAMP

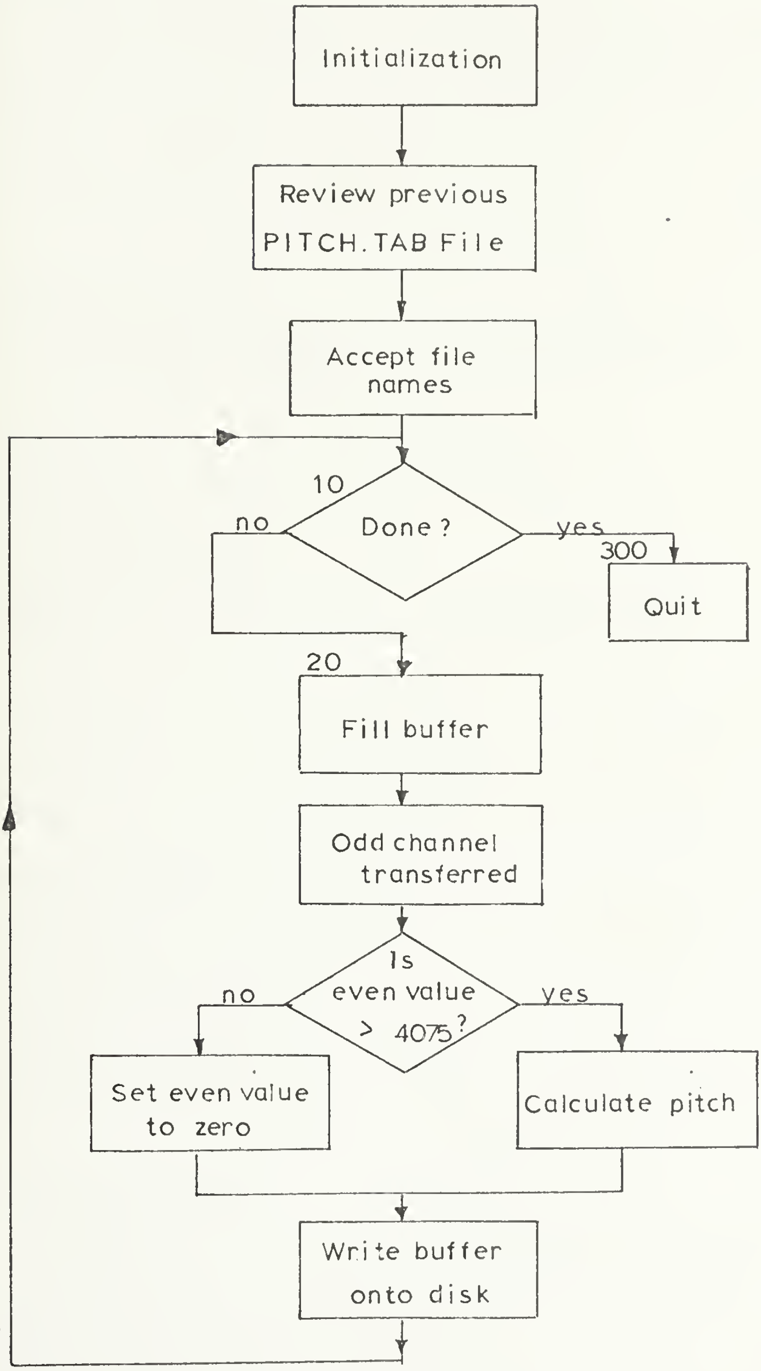


FIGURE A-9

```
C
C       TITLE:  REVAMP.FOR      750130
C
C SUBROUTINES USED:
C ISSW1 - THIS SUBROUTINE IS WRITTEN IN ASSEMBLY
C LANGUAGE AND ALLOWS FOR THE DISPLAY REGISTER SWITCHES TO
C BE USED AS PROGRAM CONTROLS.
C
C FOR SWITCH (1) UP
C INPUT AND OUTPUT FILES MAY BE SPECIFIED.
C
C FOR SWITCH (0) DOWN
C INPUT - SIGNAL.DAS
C OUTPUTS - 1.  SIGNAL.DAT  2.  PITCH.TAB.
C
C THIS PROGRAM IS USED TO REVAMP A SIGNAL FILE THAT HAS
C BEEN EDITED.  THE ODD VALUES OF THE SIGNAL ARE PRESERVED
C IN THIS TRANSFORMATION. THE EVEN VALUES OR PITCH MARKS ARE
C ZEROES UNLESS THEY ARE GE TO 4095.  AT THE SAME TIME
C THE DISTANCE BETWEEN PITCH MARKS IS MEASURED AND
C STORED IN PITCH.TAB FOR USAGE IN OTHER PROGRAMS.
C
C       INITIALIZATION
        INTEGER DATBUF(2560),BATBUF(2560),START,A(256)
        DATA START,A/2,256 * 0/
        DATA IVAR,JVAR,KVAR/2,2,2/
```

```
C REVIEW PREVIOUS PITCH.TAB FILE
      CALL ASSIGN (2,'RK1:PITCH.TAB')
      DEFINE FILE 2 (1000,2,U,JVAR)
1     READ (2,JVAR,END = 2) M, IREC
      IF (M.LT.0) A(IREC) =1
      GO TO 1
2     ENDFILE 2
C NOW REINITIALIZE.
      WRITE (7,3)
3     FORMAT (' ','SWITCH(0) DOWN - DEFAULT FILES USED.')
      WRITE (7,4)
4     FORMAT(' ','INPUT - SIGNAL.DAS; OUTPUT - SIGNAL.DAT')
      WRITE (7,5)
5     FORMAT (' ','SWITCH(0) UP - I/O FILES MAY BE CHOSEN')
      PAUSE 'PAUSE'
      IF (ISSWI(0).EQ.1) GO TO 6
      CALL ASSIGN (2,'RK1:SIGNAL.DAS')
      CALL ASSIGN (3,'RK1:SIGNAL.DAT')
      CALL ASSIGN (4,'RK1:PITCH.TAB')
      GO TO 10
6     WRITE (7,7)
7     FORMAT (' ','NAME OF INPUT FILE <FILNAM.EXT>')
      WRITE (7,8)
8     FORMAT (' ',' ')
      CALL ASSIGN (2,'RK1:SIGNAL.DAS',-14)
      WRITE (7,9)
```

```
9       FORMAT (' ','NAME OF OUTPUT FILE <FILNAM.EXT>')
        WRITE (7,8)
        CALL ASSIGN (3,'RK1:SIGNAL.DAT',-14)
        CALL ASSIGN (4,'RK1:PITCH.TAB')
10      DEFINE FILE 2 (251,256,U,IVAR)
        DEFINE FILE 3 (251,256,U,JVAR)
        DEFINE FILE 4 (1000,2,U,KVAR)
        JVAR = 2
C ARE WE THRU?
11      IF (IVAR.EQ.252) GO TO 300
C FILL THE BUFFER WITH DATA FROM INPUT.
        DO 20 I = 1,10
        READ(2'IVAR) (DATBUF(J + ((I-1)*256)),J=1,256)
20      CONTINUE
C TRANSFER THE ODD SAMPLES AS THEY ARE.  TRANSFER THE EVEN
C SAMPLES AS ZERO UNLESS THEY EQUAL 4095.  ALSO THE
C PITCH.TAB FILE IS BEING WRITTEN OUT IN THIS LOOP.
        DO 50 I1 = 1,2559,2
        BATBUF(I1) = DATBUF(I1)
        IF (DATBUF(I1+1).GE.4075) GO TO 25
        BATBUF(I1 + 1) = 0
        GO TO 50
25      BATBUF(I1 + 1) = 4095
        N = ((I1+1) - START)/2
        JU = (I1 + 1)/256
        IF((JU*256).EQ.(I1+1)) JU = JU -1
```

```
        JREC = JU + JVAR !CALCULATE REC. NO.
C CHECK FOR SILENT AREAS
        IF (M.LT.200) GO TO 40
        IF (A(JREC).NE.1) GO TO 40
        M = M * -1                     !AREA IS A SILENT ONE.
40      WRITE (4'KVAR) M, JREC
        START = I1 + 1
50      CONTINUE
C WRITE THE OUTPUT INTO SIGNAL.DAT FILE
        DO 70 I2 = 1,10
        WRITE(3'JVAR) (BATBUF(I3 + ((I2-1)*256)),I3=1,256)
70      CONTINUE
C RESET START
        START = START - 2560
        GO TO 11
300     ENDFILE 2 !TURN THINGS OFF.
        ENDFILE 3
        ENDFILE 4
        STOP
        END
```

TITLE: SHIELD.DOC

DOCUMENTATION FOR SHV03 AND SHV04

ABSTRACT

THESE PROGRAMS IMPLEMENT THE SYSTEMS DESIGNED AND FORMULATED BY VADEN SHIELDS IN AN MIT MASTER'S THESIS FROM 1970. THE PURPOSE OF THESE SYSTEMS WAS TO SEPARATE A SIGNAL FROM NOISE (USUALLY ANOTHER SPEAKER). THERE ARE TWO SYSTEMS IMPLEMENTED, AND THESE DIFFER IN THE MANNER OF UNVOICED SPEECH PROCESSING. SHV03 USES THE ATTENUATED INPUT METHOD WHILE SHV04 USES THE INERTIAL FILTERING METHOD.

SUBROUTINES USED

1. WINDOW (NTYPE,K) - THIS SUBROUTINE ALLOWS FOR THE USER TO PRESCRIBE THE WINDOW FUNCTION DESIRED IN THE PROGRAM AND TO DESIGNATE THE VALUE OF K FOR THE FILTER LENGTH.

2. COEF (NTYPE,K,A,L) - THIS SUBROUTINE CALCULATES THE COEFFICIENTS OF THE FILTER PRESCRIBED IN THE WINDOW SUBROUTINE AND STORES THESE VALUES IN AN ARRAY "A" THAT CAN BE OF MAXIMUM DIMENSION OF 15.

OTHER PROGRAMS USED IN CONJUNCTION WITH SHIELD.

1. RESAMP - THIS PROGRAM SAMPLES THE ANALOG WAVEFORM AT 10 KHZ. ON EACH OF 2 CHANNELS (4 AND 14) AND PLACES ITS OUTPUT ON FILE SIGNAL.XXX . THE CHANNELS ARE STORED IN INTERLEAVING FORMAT.

CHANNEL (4) - ODD SAMPLE VALUES. (ODD FORMAT)

CHANNEL (10) - EVEN SAMPLE VALUES. (EVEN FORMAT)

2. SEPAR - THIS PROGRAM SEPARATES THE TWO CHANNELS ON ONE FILE AND PLACES EACH ON A SEPARATE FILE.

INPUT - SIGNAL.XXX.

OUTPUTS - CHANNEL (4) - SIGNAL.SSS

CHANNEL (14) - PITCH.SSS

BOTH ARE ASSIGNED THE FORMAT OF A CHANNEL (4) ONLY FILE. (I.E. ODD SAMPLES - DATA; EVEN SAMPLES = 0 BY DEFAULT.) IF DESIRED THE SECOND OUTPUT CAN BE PUT INTO EVEN FORMAT.

3. PITV02 - THIS PROGRAM DETECTS PITCH FROM THE PITCH.SSS FILE. THERE ARE TWO OUTPUT FILES FROM THIS PROGRAM. FIRST, PITCH.TAB IS A TABULAR FILE OF THE VALUES OF THE PITCH PERIOD. THIS IS USED AS AN INPUT TO SHIELD. SECOND, PITCH.DAT IS A FILE THAT IS USED TO MARK PITCH PERIODS IN A WAVEFORM FILE. PITCH.DAT HAS TWO VALUES ONLY, (0 OR 4095). THE VALUE 4095 IS USED TO DENOTE AN ENDPOINT

OF A PITCH PERIOD. THIS IS USED AS AN INPUT TO ADDNS. PITCH. DAT IS IN EVEN FORMAT.

4. ADDNS - THIS PROGRAM ADDS TOGETHER TWO FILES, SIGNAL.XXX AND PITCH.DAT. SINCE ONE FILE, SIGNAL.XXX, IS IN ODD FOMAT AND PITCH.DAT IS IN EVEN, THE CHANNELS ARE KEPT SEPARATE BUT PUT INTO ONE FILE CALLED SIGNAL.DAT.

INPUTS TO SHIELD

1. SIGNAL.DAT - FROM ADDNS

2. PITCH.TAB - FROM PITV02

OUTPUTS FROM SHIELD

1. OUTPUT.DAT

ALGORITHM

1. THE PROGRAM FIRST PROVIDES THE USER WITH THE CHOICE OF TYPE OF WINDOW AND THE PARAMETER K. (2K + 1 COEFFICIENTS ARE USED IN THE FILTER).

2. THE PROGRAM THEN CALCULATES THE COEFFICIENTS FOR THE FILTER AND STORES THEM IN ARRAY "A".

3. THE NEXT SECTION ALLOWS FOR THE INITIALIZATION OF THE LOGICAL UNITS THAT WILL BE USED IN THIS IMPLEMENTATION.

A. LOGICAL UNIT 2 - USED FOR THE SIGNAL FILE.

B. LOGICAL UNIT 3 - USED FOR THE PITCH FILE.

C. LOGICAL UNIT 4 - USED FOR THE OUTPUT FILE.

NOTE: THE INPUT AND OUTPUT SIGNAL FILES, ON LOGICAL UNITS 2 AND 4 ARE DIVIDED INTO RECORDS OF LENGTH 256. THE PITCH FILE IS DIVIDED INTO RECORDS OF LENGTH TWO.

4.THE BUFFER IS THEN SET UP INITIALLY BY LEAVING IR RECORDS FULL OF ZEROS (IR IS COMPUTED IN THE FOLLOWING MANNER: (1/2 OF THE WINDOW LENGTH) IN SAMPLES * 1/256 (RECORD/SAMPLES) + 1 IFF THE NUMBER OF SAMPLES IS NOT A MULTIPLE OF 256.) THEN THE SIGNAL IS TAKEN FROM THE FILE AND PLACED INTO THE BUFFER. THE NUMBER OF RECORDS READ IS VARIABLE DEPENDING ON THE FIRST VALUE OF PITCH AND ON K.

5. IN THIS PROGRAM THE VALUE I IS USED AS A BUFFER POINTER. THE VALUE I POINTS TO THE CENTER OF THE FILTER THAT IS BEING USED. I IS NOW SET UP TO BE THE FIRST SAMPLE IN THE DATA SECTION OF THE BUFFER.

6. THE VALUE OF PITCH IS READ AND THROUGH THE SAME

SERIES OF MEASUREMENTS AS FOUND IN THE INITIALIZAITION PROCESS. THE ALGORITHM CHECKS TO SEE IF THE EXTREME VALUES NEEDED IN THE DIFFERENCE EQUATION ARE WITHIN THE BUFFER.

7. IF THEY ARE NOT, THE BUFFER IS SHIFTED; THE VALUE OF I REPOSITIONED AND NEW DATA IS READ IN FORM THE FILE.

8. NOW THE CALCULATIONS CAN BE PERFOMED. THE DIFFERENCE EQUATION IS IMPLEMENTED AND AN OUTPUT POINT Y IS CALCULATED AND WRITTEN ONTO THE OUTPUT FILE.

9. THE PROCESS CONTINUES UNTIL THE PITCH FLAG IS SET BY READING THE MARK IN THE INPUT FILE. A NEW VALUE OF PITCH PERIOD IS THEN READ, AND THE FILTER IS MODIFIED FOR THIS CHANGE.

10. A SLIDE OPERATION OCCURS WHEN THE VALUE OF I IS TOO SMALL TO ACCUMMODATE THE CURRENT FILTER. THE BUFFER IS REPOSITIONED AND I RESET SO THAT ALL INPUT VALUES NEEDED BY THE FILTER ARE PRESENTLY IN THE BUFFER.

11. THE PROCESSING CONTINUES UNTIL A SILENT AREA IS DETERMINED, AND THEN AFTER FILING THE OUTPUT FILE WITH ZEROS THE PROGRAM HALTS.

SHVO3

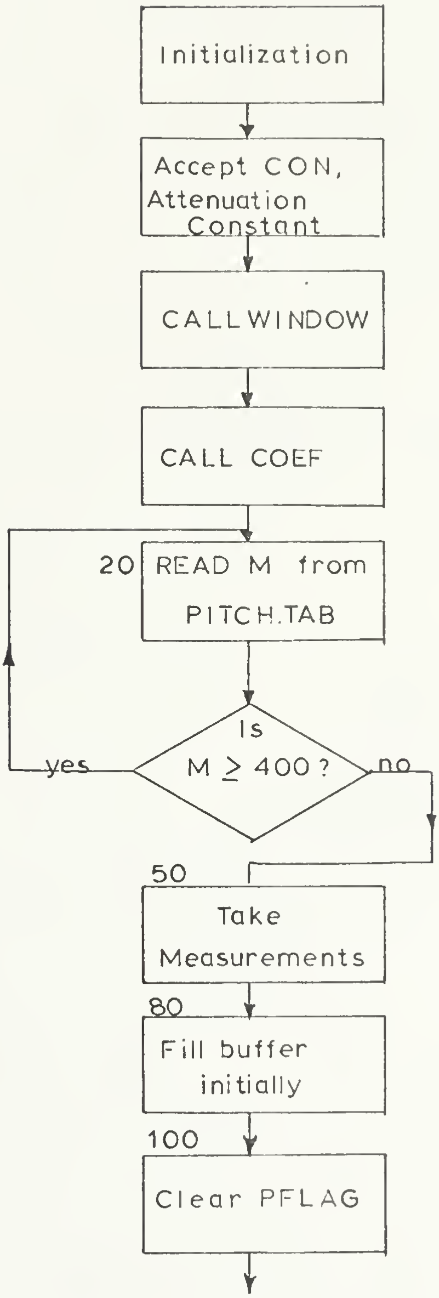


FIGURE A-10

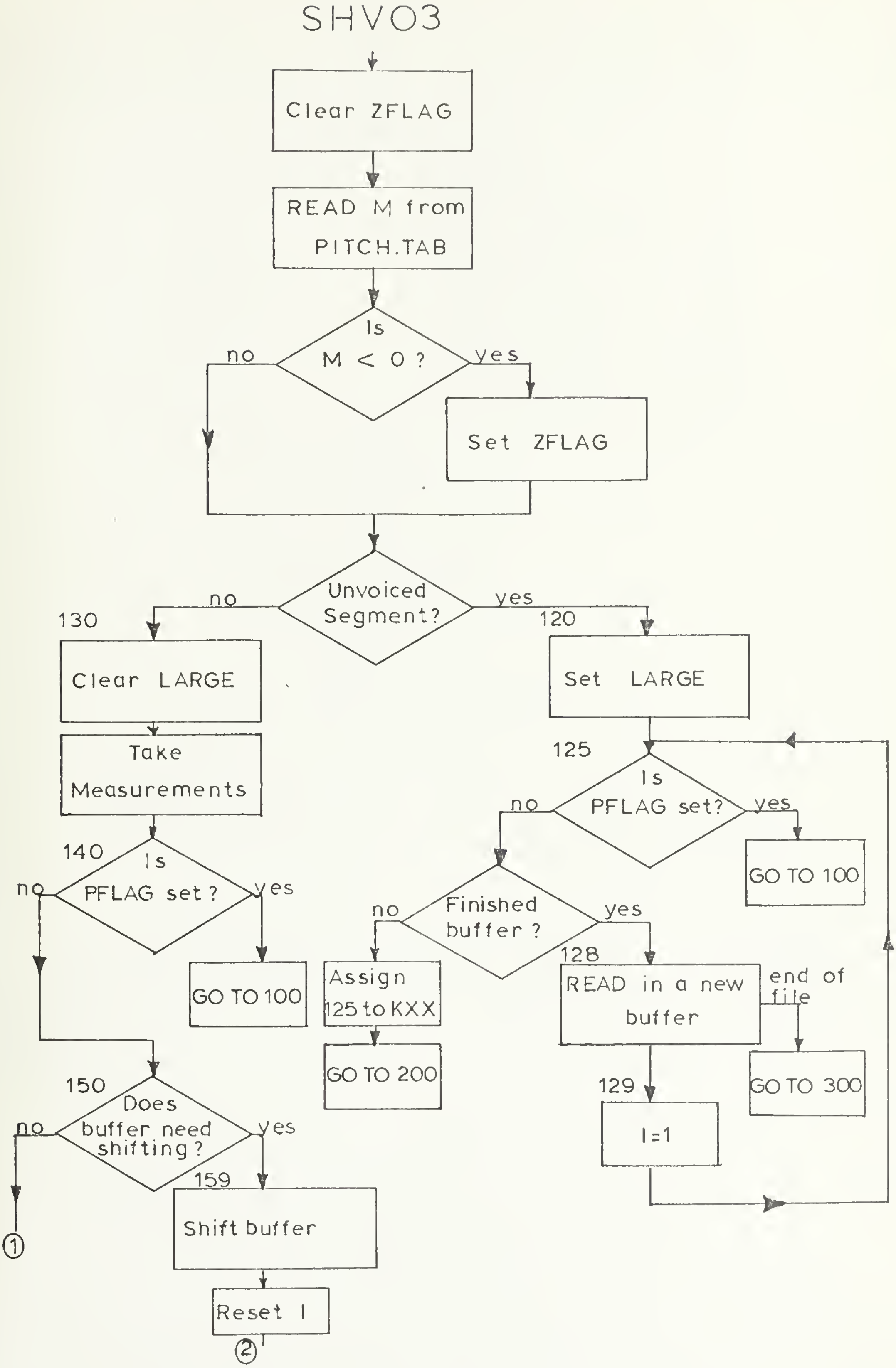


FIGURE A-10 (CON'T.)

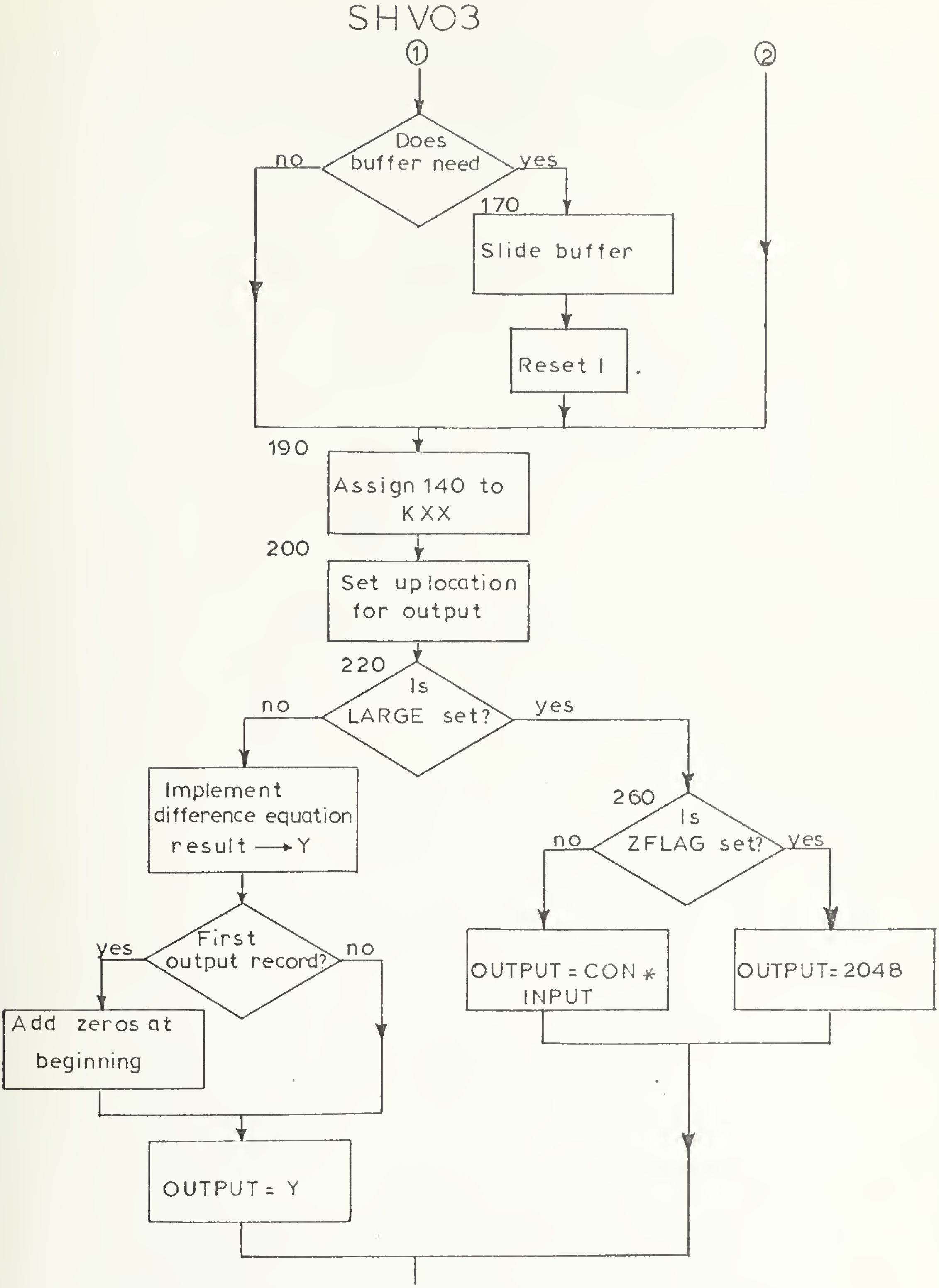


FIGURE A-10 (CON'T.)

SHV03

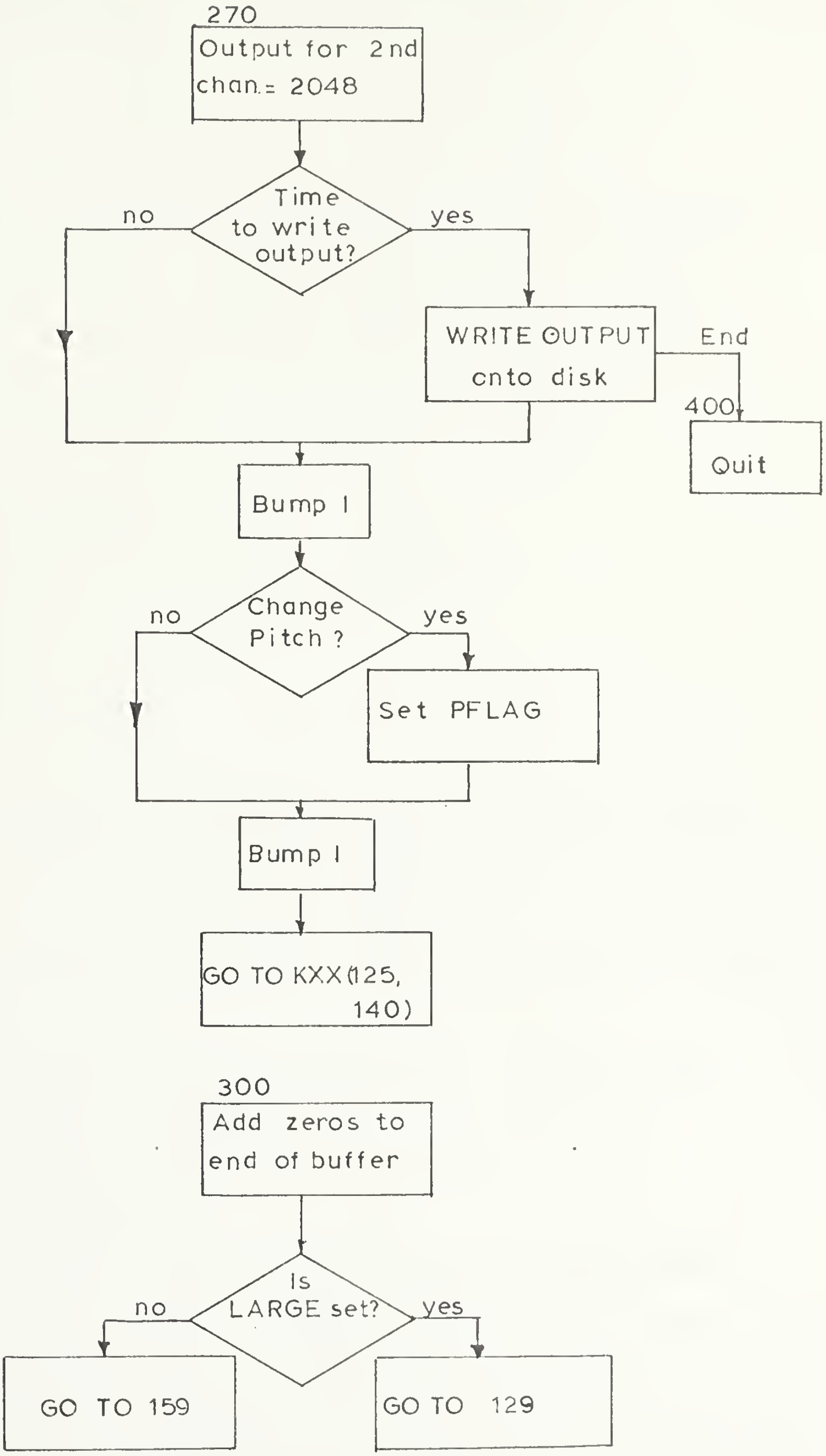


FIGURE A-10 (CON'T.)

```
C
C
C      TITLE: SHV03     AUTHOR: R. FRAZIER
C                 ADAPTED FROM THESIS BY V. SHIELDS
C
C THIS PROGRAM USES THE METHOD OF SHIELDS TO PERFORM
C THE SEPARATION OF A SPEAKER AND NOISE BY MEANS
C OF A DIGITAL COMB FILTER.  THIS PROGRAM USES THE
C ATTENUATION METHOD FOR UNVOICED SECTIONS.
C
C SUBROUTINES USED:
C WINDOW (NTYPE,K)
C COEF (NTYPE,K,A,L)
C PARAMETERS USED:
C ARRAYS:
C      1.  DATBUF - CONTAINS THE INPUT SAMPLE VALUES.
C      2.  A - CONTAINS THE COEFFICIENTS.
C      3.  OUTPUT - TEMPORARY BUFFER FOR OUTPUT VALUES.
C VARIABLES:
C      1.  SFLAG - USED TO INITIALIZE OUTPUT FILE
C          SO THAT STEP DISCONTINUITY DOES NOT
C          OCCUR.
C      2.  ZFLAG - USED TO DENOT BEG. OF SENTENCE.
C      3.  PFLAG -USED TO DENOTE NEW PITCH PERIOD.
C      4.  LARGE - USED TO DENOTE UNVOICED SECTIONS.
C      5.  CON - THE ATTENUATION CONSTANT FOR UNVOICED
```

```
C           SECTIONS.
C
C       INITIALIZATION
C
        DIMENSION A(15)
        INTEGER DATBUF(4096),START,OUTPUT(256),ZFLAG,SFLAG
        LOGICAL PFLAG,LARGE
        DATA DATBUF,SFLAG,OUTPUT/4096*0,1,256*0/
        DATA IVAR,JVAR,KVAR/2,2,2/
C CALL THE FOLLOWING SUBROUTINES FOR INITIALIZATION.
        WRITE(7,1)
1       FORMAT(' ','VALUE OF ATTEN. CONSTANT? <1,< F3.2>')
        ACCEPT 2, CON
2       FORMAT(F3.2)
        CALL ASSIGN(2,'RK1:SIGNAL.DAT')!OUTPUT FROM ADDNS.
        CALL ASSIGN(3,'PITCH.TAB',9)   !OUTPUT FROM PITDET.
        CALL ASSIGN(4,'OUTPUT',6)      !OUTPUT OF THE COMB
                                       !FILTER.
        DEFINE FILE 2 (251,256,U,KVAR)
        DEFINE FILE 3 (1000,2,U,IVAR)
C WINDOW DETERMINATION.
        CALL WINDOW (NTYPE,K)
C CALCULATE COEFFICIENTS.
        CALL COEF (NTYPE,K,A,L)
C SPECIFY THE LENGTH OF THE OUTPUT FILE.
        LIMIT = 251 + (K * 2) + 1
```

```
        DEFINE FILE 4 (LIMIT,256,U,JVAR)
C FILL THE BUFFER INITIALLY
C READ THE FIRST VALUE FROM THE PITCH TABLE FOR
C INITIALIZATION PURPOSES.
20      READ (3'IVAR) M
C DOUBLE THIS VALUE TO COMPENSATE FOR TWO CHAN.
        M=IABS(2*M)
C IS M TOO LARGE?
        IF (M.GE.400) GO TO 20
C NOW WE MAY SET UP BUFFER, TAKE MEASUREMENTS.
50      C=L-1
        IV=((C/2) * M) + 1
        IR=IV/256
        IF (IR*256.NE.IV) IR=IR + 1
        IP=IR*256*2
        I=(IP + 1) - (K * M)
C FILL THE BUFFER FROM WHERE THE POINTER STARTS.
        DO 80 II=(2*IR)+1,16
        READ(2'KVAR) (DATBUF(J + ((II-1) *256)), J=1,256)
80      CONTINUE
        IVAR=2              !RESET PITCH.TAB FILE.
C NOW BEGIN THE PROCESSING.
100     FFLAG = .FALSE. !THIS FLAG DET. WHEN WE CHANGE PITCH.
        ZFLAG = 0
        READ (3'IVAR) M
        IF (M.LT.0) ZFLAG = 1
```

```
      M=IABS(2*M)
C IS M TOO LARGE?
      IF (M.LT.400) GO TO 130 !M>400 CORRES. TO PITCH<50.
C YES, TURN OFF FILTER.
120   LARGE = .TRUE.
125   IF(PFLAG) GO TO 100     !IS PFLAG SET?
C PFLAG IS NOT SET.
C ARE WE THROUGH WITH THE BUFFER?
      IF (J.GT.4096) GO TO 127
C NO, GO RETURN AN OUTPUT POINT
      ASSIGN 125 TO KXX
      GO TO 200
C YES, WE ARE THRU BUFFER, READ IN A NEW BUFFER.
127   DO 128 JJ = 1,16
      IF(KVAR.GE.252) GO TO 300
      READ(2'KVAR) (DATBUF(ICB + ((JJ-1) * 256)),ICB=1,256)
128   CONTINUE
129   J = 1
      GO TO 125
C M WAS NOT TOO LARGE, PROCESS THE SIGNAL.
130   LARGE = .FALSE.
C TAKE MEASUREMENTS.
      IV=((C/2)  * M) + 1     !IV=1/2(LENGTH OF THE WINDOW)
      IR=IV/256
      IF (IR*256.NE.IV) IR=IR + 1
      IP=IR*256
```

```
C IS THE PITCH FLAG SET?
140     IF(PFLAG) GO TO 100
C NO, DOES THE BUFFER NEED TO BE SHIFTED?
150     IF ((I + IV).LT.4096) GO TO 160
C YES, SHIFT THE BUFFER
        IHOLD=I
        KVAR=KVAR-(2 * IR)
        DO 158 JJ=1,16
        IF (KVAR.GE.252) GO TO 300      !ARE WE THRU?
        READ (2'KVAR) (DATBUF(J + ((JJ-1)*256)),J=1,256)
158     CONTINUE
C RESET THE POINTER I.
159     START = (16 -(2 * IR)) * 256
        I = IHOLD - START
        GO TO 190
C IS I TOO SMALL?
160     IF(I.GE.IV) GO TO 190
C YES, SLIDE THE BUFFER.
170     JHOLD = I
        JVEX = IV - I
        JVEY = JVEX/256
        IF((JVEY * 256).NE.JVEX) JVEY = JVEY + 1
        KVAR = (KVAR - 16) - JVEY
        DO 175 IDY = 1,16
        READ(2'KVAR) (DATBUF( IDX + ((IDY-1)*256)),IDX=1,256)
175     CONTINUE
```

```
C RESET I
      I = JHOLD + (256 * JKEY)
190   ASSIGN 140 TO KXX
C COMPUTATION AND OUTPUT SECTION
200   IND = I
210   IF(IND.LE.255) GO TO 220
      IND = IND - 256
      GO TO 210
220   IF (LARGE) GO TO 260    !LARGE SET?
C NO, CALCULATE AND OUTPUT POINT.
      Y=0.0
      DO 240 MM=1,K + 1
      IPOINT=I - ((K + 1 - MM) *  M)
      Y=Y + (A(MM) * DATBUF(IPOINT))
240   CONTINUE
C IS THIS THE FIRST OUTPUT TO BE WRITTEN.
      IF (SFLAG.EQ.0) GO TO 255
C YES, DON'T ALLOW DRASTIC STEP INPUT.
      DO 254 IU = 1,(IND-2),2
      OUTPUT (IU) = 2048
      OUTPUT (IU + 1) = 0
254   CONTINUE
      SFLAG = 0 !FOREVER.
C WRITE THE OUTPUT POINT.
255   OUTPUT(IND) = Y
      GO TO 270
```

```
260    IF(ZFLAG.EQ.0) GO TO 265
       OUTPUT(IND) = 2048
       GO TO 270
265    OUTPUT(IND) = (CON * (DATBUF(I) - 2048)) + 2048
270    OUTPUT(IND + 1) = 0
C ARE WE READY TO DUMP OUTPUT?
       IF(IND.NE.255) GO TO 280
C YES,WE ARE!
       WRITE(4'JVAR,END=400) (OUTPUT(KKK),KKK=1,256)
280    I=I + 1
       IF(DATBUF(I).EQ.4095) PFLAG=1   !CHANGE PITCH?
       I=I + 1
290    GO TO KXX, (125,140)
C THIS SECTION IS USED TO FILL DATBUF WITH ZEROS WHEN
C CALLED.
300    DO 310 ILAX=JJ,16         !DETERMINE HOW MANY
       DO 305 JLAX=1,256
       DATBUF(JLAX + ((ILAX-1)*256))=0        !RECORDS NEED
 TO BE
305    CONTINUE                  !ADDED WITH ZEROS.
310    CONTINUE
       IF(LARGE) GO TO 129
       GO TO 159
C YES, QUIT
400    ENDFILE 2
       ENDFILE 3
```

```
ENDFILE 4
STOP
END
```

```
C
C
C       TITLE: SHV04     AUTHOR: R. FRAZIER
C                   ADAPTED FROM THESIS BY V. SHIELDS
C
C THIS PROGRAM USES THE METHOD OF SHIELDS TO PERFORM
C THE SEPARATION OF A SPEAKER AND NOISE BY MEANS
C OF A DIGITAL COMB FILTER.  THIS PROGRAM USES THE
C ATTENUAITON METHOD FOR UNVOICED SECTIONS.
C
C SUBROUTINES USED:
C WINDOW (NTYPE,K)
C COEF (NTYPE,K,A,L)
C
C PARAMETERS USED:
C ARRAYS:
C       1.  DATBUF - CONTAINS THE INPUT SAMPLE VALUES.
C       2.  A - CONTAINS THE COEFFICIENTS.
C       3.  OUTPUT - TEMPORARY BUFFER FOR OUTPUT VALUES.
C VARIABLES:
C       1.  SFLAG - USED TO INITIALIZE OUTPUT FILE
C           SO THAT STEP DISCONTINUITY DOES NOT
C           OCCUR.
C       2.  ZFLAG - USED TO DENOTE BE. OF SENTENCE.
C       3.  PFLAG - USED TO DENTOE NEW PITCH PERIOD.
C       4.  LARGE - USED TO DENOTE UNVOICED SECTIONS.
```

```
C
C     INITIALIZATION
C
      DIMENSION A(15)
      INTEGER DATBUF(4096),START,OUTPUT(256),ZFLAG,SFLAG
      LOGICAL PFLAG,LARGE
      DATA DATBUF,SFLAG,OUTPUT/4096*0,1,256*0/
      DATA IVAR,JVAR,KVAR/2,2,2/
C CALL THE FOLLOWING SUBROUTINES FOR INITIALIZATION.
      CALL ASSIGN(2,'DK1:SIGNAL.DAT')!OUTPUT FROM ADDNS.
      CALL ASSIGN(3,'PITCH.TAB',9)   !OUTPUT FROM PITV02.
      CALL ASSIGN(4,'OUTPUT',6)      !OUTPUT OF THE COMB
                                     !FILTER.
      DEFINE FILE 2 (251,256,U,KVAR)
      DEFINE FILE 3 (1000,2,U,IVAR)
C WINDOW DETERMINATION.
      CALL WINDOW(NTYPE,K)
C CALCULATE COEFFICIENTS.
      CALL COEFF(NTYPE,K,A,I)
C SPECIFY THE LENGTH OF THE OUTPUT FILE.
      LIMIT = 251 + (K * 2) + 1
      DEFINE FILE 4 (LIMIT,256,U,JVAR)
C FILL THE BUFFER INITIALLY
C READ THE FIRST VALUE FROM THE PITCH TABLE FOR
C INITIALIZATION PURPOSES.
20    READ (3'IVAR) M
```

```
C DOUBLE THIS VALUE TO COMPENSATE FOR TWO CHANNELS.
      M=IABS(2*M)
      MHOLD = M !THIS WILL BE USED FOR UNVOICED.
C IS M TOO LARGE?
      IF (M.GE.400) GO TO 20
C NOW WE MAY SET UP BUFFER, TAKE MEASUREMENTS.
50    C=L-1
      IV=((C/2) * M) + 1
      IR=IV/256
      IF (IR*256.NE.IV) IR=IR + 1
      IP=IR*256*2
      I=(IP + 1) - (K * M)
C FILL THE BUFFER FROM WHERE THE POINTER STARTS.
      DO 80 II=(2*IR)+1,16
      READ(2'KVAR) (DATBUF(J + ((II-1) *256)),J=1,256)
80    CONTINUE
      IVAR=2               !RESET PITCH.TAB FILE.
C
C NOW BEGIN THE PROCESSING.
100   FFLAG = .FALSE.  !THIS FLAG SET. WHEN WE CHANGE
 PITCH.
      ZFLAG = 0
      MHOLD = M
      READ (3'IVAR) M
      IF (M.LT.0) ZFLAG = 1
      M=IABS(2*M)
```

```
C IS M TOO LARGE?
      IF (M.GE.400) M = MHOLD
C TAKE MEASUREMENTS.
      IV=((C/2)  * M) + 1      !IV=1/2(LENGTH OF THE WINDOW)
      IR=IV/256
      IF (IR*256.NE.IV) IR=IR + 1
      IP=IR*256
C IS THE PITCH FLAG SET?
140   IF(PFLAG) GO TO 100
C NO, DOES THE BUFFER NEED TO BE SHIFTED?
150   IF ((I + IV).LT.4095) GO TO 160
C YES, SHIFT THE BUFFER
      IHOLD=I
      KVAR=KVAR-(2 * IR)
      DO 158 JJ=1,16
      IF (KVAR.GE.252) GO TO 300      !ARE WE THRU?
      READ (2'KVAR) (DATBUF(J + ((JJ-1)*256)),J=1,256)
158   CONTINUE
C RESET I
159   START=(16-(2*IR)) * 256
      I=IHOLD-START
      GO TO 200
C IS I TOO SMALL?
160   IF(I.GE.IV) GO TO 200
C YES, SLIDE THE BUFFER.
170   JHOLD = I
```

```
        JVEX = IV - 1
        JVEY = JVEY/256
        IF((JVEY * 256).NE.JVEX) JVEY = JVEY + 1
        KVAR = (KVAR - 16) - JVEY
        DO 175 IDY = 1,16
        READ(2'KVAR) (DATBUF( IDX + ((IDY-1)*256)),IDX=1,256)
175     CONTINUE
C RESET I
        I = JHOLD + (256 * JVEY)
C COMPUTAITON AND OUTPUT SECTION
200     IND = I
210     IF(IND.LE.256) GO TO 220
        IND = IND - 256
        GO TO 210
C NO, CALCULATE AND OUTPUT POINT.
220     IF (ZFLAG.EQ.1) GO TO 253
        Y=0.0
        DO 240 MM=1,K-1
        IPOINT=I - ((K + 1 - MM) *  M)
        Y=Y + (A(MM) * (DATBUF(IPOINT) - 2048))
240     CONTINUE
C IS THIS THE FIRST OUTPUT TO BE WRITTEN?
253     IF (SFLAG.EQ.0) GO TO 255
C YES, DON'T ALLOW DRASTIC STEP TRANSIENT.
        DO 254 IU = 1,(IND -2), 2
        OUTPUT (IU) = 2048
```

```
      OUTPUT (IU + 1) = 0
254   CONTINUE
      SFLAG = 0 !FOREVER
C WRITE THE OUTPUT POINT.
255   IF(ZFLAG.EQ.0) OUTPUT(IND) = IFIX(Y + .5) + 2048
      IF (ZFLAG.EQ.1) OUTPUT (IND) = 2048
270   IF (DATBUF(I + 1).GE.4075) OUTPUT(IND + 1) = 4095
      IF (DATBUF(I + 1).LT.4075) OUTPUT(IND + 1) = 0
C ARE WE READY TO DUMP OUTPUT?
      IF(IND.NE.255) GO TO 280
C YES,WE ARE!
      WRITE(4'JVAR,END=400) (OUTPUT(KKK),KKK=1,256)
280   I=I + 1
      IF(DATBUF(I).EQ.4095) PFLAG=1  !CHANGE PITCH?
      I=I + 1
290   GO TO 140
C THIS SECTION ADDS ZEROS TO DATBUF WHEN CALLED.
300   DO 310 ILAX=JJ,16
      DO 305 JLAX=1,256
      DATBUF(JLAX + ((ILAX-1)*256))=2048
305   CONTINUE
310   CONTINUE
      GO TO 159
C YES, QUIT
400   ENDFILE 2
      ENDFILE 3
```

```
ENDFILE 4
STOP
END
```

COEF

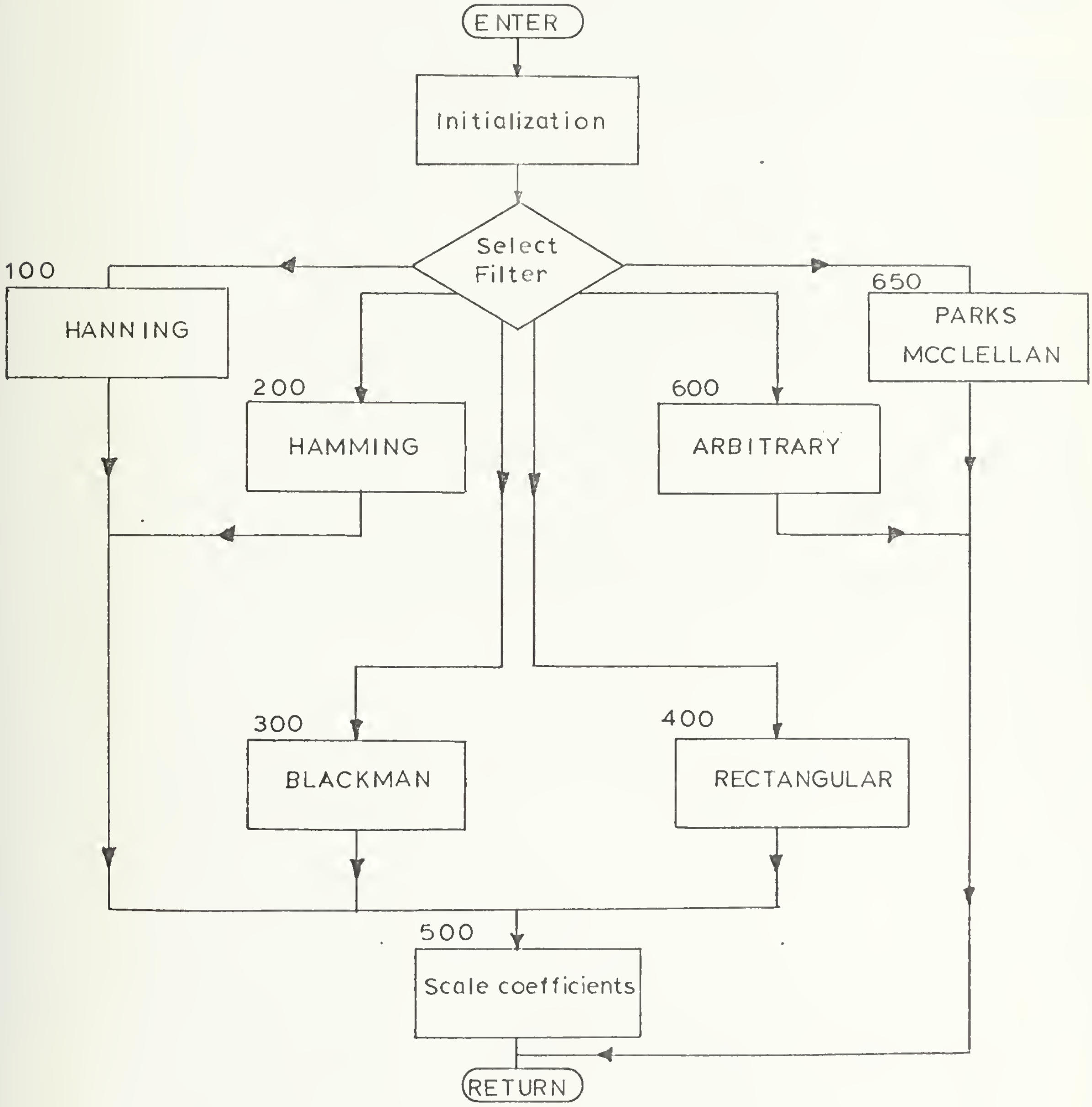


FIGURE A-11

```
C
C
C       TITLE:  COEF.FOR 741206
C
C THIS SUBROUTINE TAKES THE VALUES THAT WERE READ BY WINDOW
C AND CALCULATES THE COEFFICIENTS FOR THE APPROPRIATE
C WINDOW THAT WAS REQUESTED.  IT RETURNS THESE COEFFICIENTS
C TO THE MAIN PROGRAM IN ARRAY "A".
C DEFINITIONS:
C       1.  A - ARRAY CONTAINING COEFFICIENTS.
C       2.  NTYPE - THE VALUE THAT CORRESPONDS TO THE
C           WINDOW OR FILTER DESIRED.
C       3.  K - THE VALUE THAT DETERMINES THE LENGTH
C           OF THE IMPULSE RESPONSE OF THE WINDOW
C           AND THE NUMBER OF COEFFICIENTS CALCULATED.
C       4.  L = 2 * K + 2
C
        SUBROUTINE COEF (NTYPE,K,A,L)
        DIMENSION A(15)
        DO 10 I = 1,15
        A(I) = 0.0
10      CONTINUE
        PI = 3.14159
        SCALE = 0.0
        L = 2 * K + 2
        GO TO (100,200,300,400,600,650), NTYPE
```

```
C HANNING WINDOW SELECTED
100     DO 110 I = 1,L-1
        A(I) = .5 - .5*(COS((2*PI*I)/L))
        SCALE = SCALE + A(I)
110     CONTINUE
        GO TO 500
C HAMMING WINDOW SELECTED.
200     DO 210 I = 1,L-1
        A(I) = .54 - (.46*((COS((2*PI*I)/L)))
        SCALE = SCALE + A(I)
210     CONTINUE
        GO TO 500
C BLACKMAN WINDOW SELECTED
300     DO 310 I = 1,L-1
        A(I) = .42-(.5*(COS((2*PI*I)/L))) + (.08*(COS((4*PI*
       1 I)/L)))
        SCALE = SCALE + A(I)
310     CONTINUE
        GO TO 500
C RECTANGULAR WINDOW SELECTED.
400     DO 410 I=1,L-1
        A(I) = 1.0
        SCALE = SCALE + A(I)
410     CONTINUE
C COMPUTE THE SCALE FACTOR.
500     DO 520 J = 1,15
```

```
      A(I) = A(I)/SCALE
520   CONTINUE
      GO TO 700
C AN ARBITRARY SET OF COEFFICIENTS MAY BE TYPED IN.
600   CALL ASSIGN (5,'RK1:',-1)
      DEFINE FILE 5 (2,256,U,NVAR)
      NVAR = 1
      READ (5,NVAR) (A(IY),IY = 1,2*K+1)
      ENDFILE 5
      GO TO 700
C PARKS_MCCLELLAN COEFFICIENTS ARE READ IN FROM FILE.
650   CALL ASSIGN (5,'RK1:',-1)
      READ (5) NFIL,NEG
      NN = (NFIL + 1)/2
      READ (5) (A(IY),IY=1,NN)
      ENDFILE 5
      DO 660 J= 1,NN
      A(NFIL + 1 -J) = A (J)
660   CONTINUE
700   RETURN
      END
```

WINDOW

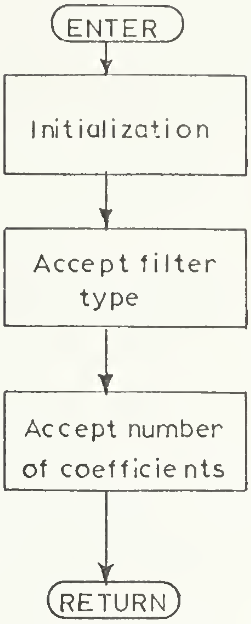


FIGURE A-12

```
C
C
C       TITLE:  WINDOW.FOR      741220
C
C THIS SUBROUTINE ALLOWS FOUR POSSIBLE WINDOWS,
C A PARKS-MCCLELLAN FILTER, OR AN ARB. FILTER TO
C BE SELECTED AND A VALUE OF K (PROPORTIONAL TO THE
C LENGTH OF THE WINDOW) TO BE SELECTED FOR THE MAIN
C PROGRAM TO USE.
C
        SUBROUTINE WINDOW (ITYPE,K)
        WRITE (7,10)
10      FORMAT(' ','TYPE IN NUMBER CORRESP. TO WINDOW',/,1X,
     1  'DESIRED: HANN=1, HAMM.=2, BLACK.=3, RECT.=4' ,/,
     2  'ARBITRARY = 5, PARKS-MCCLELLAN = 6 (I1)',/)
        READ (7,20) NTYPE
20      FORMAT(I1)
        WRITE(7,30)
30      FORMAT(' ','THE VALUE OF K=? (2K + 1= LENGTH)')
        READ (7,40) K
40      FORMAT(I2)
        RETURN
        END
C
C
```

FILIN

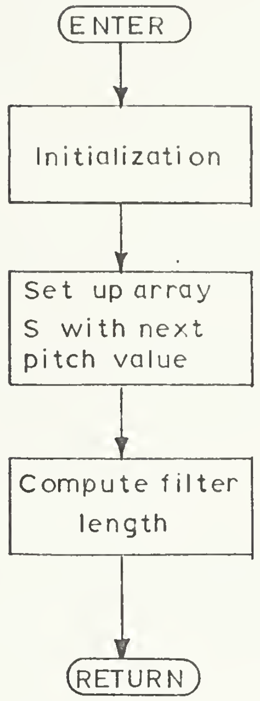


FIGURE A-13

```
C
C       TITLE:  FILIN.FOR        750315
C
C
C THIS SUBROUTINE ACTS TO INITIALIZE THE FILTER AT THE
C BEGINNING OF THE PROGRAM OR AT THE ONSET OF A VOICED
C AREA AFTER AN UNVOICED AREA.
C
C       INITIALIZATION
        SUBROUTINE FILIN (K,IPIT,S,ISUM)
        INTEGER S(14)
        DO 10 KC = 1,2*K
        S(KC) = IABS (IPIT * 2)
10      CONTINUE
C ISUM = THE LENGTH OF THE FILTER.
        ISUM = 4 * K * IPIT      !RENEW ISUM.
        RETURN
        END
```

CALC

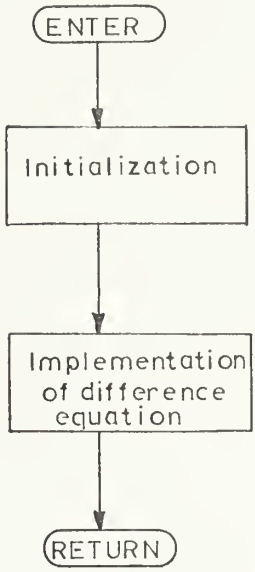


FIGURE A-14

```
C
C
C       TITLE:  CALC.FOR 750313
C
C THIS SUBROUTINE IS USED WITH THE MAIN PROGRAM NONV01
C AND IS USED TO CALCULATE AN OUTPUT POINT.
C PARAMETERS:
C ARRAYS:
C       1.  A - CONTAINS THE COEFFICIENTS
C       2.  DATBUF - CONTAINS THE INPUT DATA
C       3.  S - CONTAINS THE SPECINGS BETWEEN THE
C             COEFFICIENTS.
C VARIABLES:
C       1.  I - POINTER OF THE CURRENT PROCESSING IN DATBUF
C       2.  K - VALUE THAT DETERMINES THE NO. OF COEFF.
C       3.  IOUT - A VALUE RETURNED TO THE MAIN PROGRAM.
C             THE OUTPUT OF THE FILTER FROM ONE POINT.
C
        SUBROUTINE CALC (A,DATBUF,S,I,K,IOUT)
        DIMENSION A(15)
        INTEGER DATBUF (4096),S(14),OUT
C COMPUTATION OF THE OUTPUT POINT
        Y = 0.0 !INITIALIZE THE SUM VALUE.
        Y = A(1) * DATBUF(I)    !PERFORM FIRST MULT.
        INDEX = I
        DO 20 IY = 2,(2*K) + 1  !PERFORM REST OF MULT.
```

```
      INDEX = INDEY - S(IY -1)
      Y = Y + (A(IY)* DATBUF(INDEY))
20    CONTINUE
      IOUT = IFIX (Y + .5)     !ROUND THE OUTPUT VALUE
      RETURN
      END
```

TITLE: ADAPT.DOC

DOCUMENTATION FOR NONVO1 AND NONVO2

ABSTRACT

THESE PROGRAMS IMPLEMENT THE ADAPTIVE SYSTEMS FORMULATED IN THIS THESIS. NONVO1 CORRESPONDS TO THE ADAPTIVE OVERLOAD SYSTEM WHILE NONVO2 IS THE ADAPTIVE SYSTEM.

SUBROUTINES USED:

A. NONVO1

1. WINDOW (NTYPE,K) - THIS SUBROUTINE ALLOWS FOR THE USER TO PRESCRIBE THE WINDOW FUNCTION, PARKS-MCCLELLAN FILTER, OR AN ARBITRARY FILTER DESIRED IN THE PROGRAM AND TO DESIGNATE THE VALUE OF K FOR THE FILTER LENGTH.

2. COEF (NTYPE,K,A,L) - THIS SUBROUTINE CALCULATES THE COEFFICIENTS OF THE FILTER PRESCRIBED IN THE WINDOW SUBROUTINE AND STORES THEM IN AN ARRAY "A" THAT CAN BE OF MAXIMUM DIMENSION OF 15.

3. CALC (A,DATRUE,S,I,K,IOUT) - THIS SUBROUTINE

RETURNS ONE VALUE OF THE OUTPUT PER CALL. IT IMPLEMENTS THE ADAPTIVE FILTER DIFFERENCE EQUATION.

4. FILIN (K,NEWPI,S,ISUM) - THIS SUBROUTINE INITIALIZES THE FILTER AT THE BEGINNING OF THE PROGRAM AND AT THE ONSET OF A VOICED AREA FOLLOWING AN UNVOICED AREA. THE FILTER SPACINGS ARE ALL SET EQUAL TO THE NEW PITCH PERIOD VALUE INITIALLY.

B. VONVO2

1. WINDOW (NTYPE,K) - (SEE EXPLANATION ABOVE).

2. COEF (NTYPE,K,A,L) - (SEE EXPLANATION ABOVE).

3. FILIN (K,NEWPI,S,ISUM) - (SEE EXPLANATION ABOVE).

4. CALC2 (1,DATBUF,S,I,K,IOUT,ISTART) - THIS SUBROUTINE IS BASICALLY THE SAME AS THE CALC SUBROUTINE WITH THE EXCEPITON THAT IT PROVIDES FOR THE OVERLOAD PROBLEM AND CORRECTS FOR IT.

5. BEG (ISTART,S,I,K) - THIS SUBROUTINE INITIALIZES THE ISTART ARRAY WHICH IS USED IN THE DETECTION OF THE

OVERLOAD PROBLEM.

6. REBEG (ISTART,I,K) - THIS SUBROUTINE RE-INITIALIZES THE ISTART ARRAY AFTER A BUFFER SHIFT HAS BEEN PERFORMED.

OTHER PROGRAMS USED WITH THESE SYSTEMS

(SEE THE SHIELD.DOC EXPLANATION; ALL PROGRAMS ARE THE SAME.)

INPUTS TO ADATIVE SYSTEM

1. SIGNAL.DAT - FROM ADDNS

2. PITCH.TAB - FROM FITV02

OUTPUT FROM THE ADAPTIVE SYSTEM

1. OUTPUT.DAT

ALGORITHM

1. THE ALGORITHM IS BASICALLY THE SAME AS THE SHIELDS' SYSTEM. ONLY THE EXCEPTIONS ARE LISTED IN THIS DESCRIPTION.

2. THE PITCH.TAB FILE IS INITIALLY STORED IN AN ARRAY "P" SO EASIER ACCESS TO THE PITCH PERIOD VALUES CAN BE OBTAINED.

3. AN ARRAY "S" IS USED TO HOLD THE SPACING VALUES BETWEEN THE COEFFICIENTS. THE VARIABLE ISUM IS USED TO

DENOTE THE LENGTH OF THE FILER.

4. IN NONVO2 AN ARRAY "ISTART" IS USED TO HOLD THE INITIAL STARTING VALUES FOR A PARTICULAR FILTER BASED ON A PITCH PERIOD VALUE. IT IS USED IN DETECTING AN OVERLOAD CONDITION.

5. THE OUTPUT POINT IS COMPUTED BY THE DIFFERENCE EQUATION USED FOR THE ADAPTIVE FILTER. IN NONVO2 THE OVERLOAD PROBLEM IS CORRECTED.

6. THE BUFFER IS SHIFTED IN A SIMILIAR MANNER TO THE SHIELDS' PROGRAM.

7. DURING UNVOICED SECTIONS THE INPUT IS ATTENUATED BY THE VARIABLE, CON, TO OBTAIN THE OUTPUT.

8. WHEN THE LOGICAL VARIABLE, DONE, IS TRUE, THE PROGRAM PERFORMS A SERIES OF CLOSING STEPS, AND THEN HALTS.

END

```
C
C
C
C      TITLE:  NONVO1.FOR      750311
C
C
C THIS PROGRAM IMPLEMENTS THE NONUNIFORMLY SPACED
C ADAPTIVE FILTER WITHOUT CORRECTING FOR THE OVERLOAD
C PROBLEMS.
C
C SUBROUTINES USED:
C      1.  WINDOW (NTYPE,K)
C      2.  COEF (NTYPE,K,A,L)
C      3.  FILIN (K,NEWP1,S,ISUM)
C      4.  CALC (A,DATBUF,S,I,K,IOUT)
C
C ARRAYS:
C      1.  A - CONTAINS COEFFICIENTS.
C      2.  DATBUF - INPUT VALUES.
C      3.  OUTPUT - TEMPORARY OUTPUT ARRAY.
C      4.  P - USED TO STORE PITCH PERIODS.
C      5.  S - USED TO STORE SPACINGS BETWEEN COEF.
C
C VARIABLES:
C      1.  DONE - LOGICAL DENOTES END OF SENTENCE.
C      2.  ATTEN - DENOTES UNVOICED AREAS.
```

```
C       3.  J - POINTER IN PITCH PERIOD ARRAY "P".
C       4.  ISUM - LENGTH OF THE FILTER.
C
C       INITIALIZATION
        DIMENSION A(15)
        INTEGER DATBUF(4096),OUTPUT(256),P(1000),S(14)
        LOGICAL DONE, ATTEN
        DATA DATBUF,OUTPUT,P,S/4095*0,256*0,1000*0,14*0/
        DATA IVAR,JVAR,KVAR/2,2,2/
        DONE = .FALSE.
        ATTEN = .FALSE.
        CALL ASSIGN (2,'RK1:SIGNAL.DAT')
        CALL ASSIGN (3,'RK1:PITCH.TAB')
        CALL ASSIGN (4,'RK1:OUTPUT.DAT')
        DEFINE FILE 2 (251,256,U,IVAR)
        DEFINE FILE 3 (1000,2,U,JVAR)
        CON = .3  !ATTENUATION CONSTANT
C CHOOSE WINDOW TYPE
        CALL WINDOW (NTYPE,K)
C CALCULATE THE COEFFICIENTS
        CALL COEF (NTYPE,K,A,L)
C SPECIFY THE LENGTH OF THE OUTPUT FILE.
        LIMIT = 251 + (K*2) + 1
        DEFINE FILE 4 (LIMIT,256,U,KVAR)
C FILL THE PITCH ARRAY.
        DO 100 IA = 1,1000
```

```
        READ (3'JVAR,END = 110) P(JA)
100     CONTINUE
110     CONTINUE
        J = 1 !J IS THE POINTER IN THE PITCH PEROD ARRAY.
C THIS PART DETERMINES IF WE HAVE A SILENT AREA AT
C THE BEGINNING OF THE FILE.
120     IF (P(J).GE.0) GO TO 200 !SILENT AREA?
        M = IABS(P(J) * 2) !YES, SET THE OUTPUT = 0.
        DO 150 KD = 1,(M-1),2
        KA = KD
130     IF (KA.LE.255) GO TO 140
        KA = KA - 256
        GO TO 130
140     OUTPUT (KA) = 2048
        OUTPUT(KA + 1) = 0
        IF (KA.NE.255)GO TO 150 !IS IT TIME TO WRITE A REC.?
        WRITE(4'KVAR) (OUTPUT(KB),KB=1,256)    !YES.
150     CONTINUE
C PUMP THE COUNTER.
        J = J + 1
        GO TO 120
C PITCH PERIOD WAS NOT A SILENT ONE.
200     IF(P(J).GE.400) GO TO 3000
C VOICED, SET UP THE FILTER SPACING.
        ASSIGN 350 TO KXX
        CALL FILIN (K,P(J),S,ISUM)
```

```
C MEASUREMENTS SECTION
      IR = ISUM /256
      IF ((IR * 256).NE.ISUM) IR = IR + 1
C SET UP FOR INITIAL READING INTO BUFFER.
      IVAR = KVAR - IR
      DO 300 KE = 1,16
      READ(2'IVAR) (DATBUF(KF + ((KE-1)*256)),KF = 1,256)
300   CONTINUE
C SET UP THE BUFFER POINTER
      IF (KA.EQ.255) KA = -1    !RESET THE POINTER
                                !TO THE LAST OUTPUT.
      I = (IR * 256) + (KA + 2)
C DO THE OUTPUT CALCULATIONS
350   CALL CALC (A,DATBUF,S,I,K,IOUT)
C COMPUTE THE ADDRESS FOR THE OUTPUT POINT.
360   IND = 1
365   IF (IND.LE.255) GO TO 370 !DO WE HAVE A GOOD ADDRESS
      IND = IND - 256                     !NO GO RECHECK.
      GO TO 365
370   IF(ATTEN) GO TO 375       !IS THIS A VOICED SEGMENT?
      OUTPUT(IND) = IOUT !YES, GET CALCULATED VALUE.
      GO TO 380
C ATTENUATE THE INPUT.
375   OUTPUT(IND)=(CON*(DATBUF(I)-2048))+2048
380   OUTPUT(IND+1) = 0         !SECOND CHANNEL = 0.
C IS IT TIME TO DUMP THE OUTPUT BUFFER?
```

```
      IF (IND.NE.255) GO TO 400
      WRITE(4'KVAR,END=3000) (OUTPUT(KB),KB=1,256) !YES.
400   I = I + 1
C IS THERE A PITCH MARK?
      IF (DATBUF(I).LT.4075) GO TO 1220
500   NEWPI = P(J) * 2 !YES, GET A NEW VALUE OF PITCH.
      J = J + 1 !BUMP THE PITCH PERIOD POINTER.
C ARE WE ENTERING A SILENT AREA?
      IF (NEWPI.LT.0) GO TO 1600
C NO, ARE WE ENTERING AN UNVOICED AREA?
      IF (NEWPI.GT.400) GO TO 1800
C NO, THEREFORE WE ARE ENTERING A VOICED AREA.
C CHANGE THE FILTER.
      ASSIGN 350 TO KXX
C IS THE FILTER BEING INITIALIZED AFTER AN UNVOICED AREA?
      IF (ATTEN) GO TO 800
      GO TO 900 !THE FILTER IS NOT BEING RE-INITIALIZED.
800   ATTEN = .FALSE.  !CLEAR THE FLAG.
      CALL FILIN (K,NEWPI,S,ISUM)
      GO TO 1220
900   KAA = 2 * K
1000  IF(KAA.EQ.1) GO TO 1100
      S(KAA) = S(KAA - 1) !SHIFT THE SPACINGS
      KAA = KAA -1
      GO TO 1000
1100  S(1) = NEWPI !SET THE NEW PITCH PERIOD SPACING
```

```
        ISUM = 0 !COMPUTE THE LENGTH OF THE FILTER
        DO 1200 K1 = 1,2*K
        ISUM = ISUM + S(K1)
1200    CONTINUE
1220    I = I + 1
C IS IT TIME TO SHIFT THE BUFFER?
        IF (I.LT.4096) GO TO 1400
C YES, SHIFT THE BUFFER.
        IR = ISUM / 256
        IF ((IR*256).NE.ISUM) IR = IR + 1
        IVAR = IVAR - IR
        DO 1300 K2 = 1,16
        IF (IVAR.GE.252) GO TO 2900
        READ (2'IVAR) (DATBUF(K3 + ((K2-1)*256)),K3 = 1,256)
1300    CONTINUE
C RESET I
1310    I = (IR * 256) + 1
1400    GO TO KXX, (350,360,1840)
C THIS AREA RESERVED FOR SILENT AREA LATER.
1600    NEWPI = IABS(NEWPI) !GET RID OF MINUS SIGN
        DONE = .TRUE.            !SET DONE FLAG.
        GO TO 1800
C UNVOICED AREA
1800    I = I + 1           !BUMP THE POINTER.
        IF (NEWPI.LE.ISUM) GO TO 2000
1820    K7 = 1              !SET UP A COUNTER.
```

```
1825    KAB = 0
C ARE WE THROUGH SHIFTING?
1830    IF (KAB.EQ.((2*K)-K7))GO TO 1835
        KNEW = 2 * K - KAB
        KOLD = KNEW - 1
        S(KNEW) = S(KOLD)         !NO,CONTINUE
        KAB = KAB + 1
        GO TO 1830
1835    K8 = 1
1837    ASSIGN 1840 TO KXX
        GO TO 350 !GO COMPUTE AN OUTPUT.
1840    K8 = K8 + 2
        IF (K8.LT.S(1)) GO TO 1837      !ARE WE THRU?
        K7 = K7 + 1                     !YES, BUMP K7
C ENTER THE ATTENUATION PROCEDURE
1850    ATTEN = .TRUE.            !SET THE ATTENUATION FLAG.
        IF (DONE) GO TO 1900      !ARE WE ALMOST THROUGH?
        ASSIGN 360 TO KXX !NO, THIS IS ONLY AN UNVOICED AREA.
        GO TO 360
1900    IND = 1
C DO WE HAVE A SUITABLE IND?
1920    IF (IND.LE.255) GO TO 1950
        IND = IND - 256                 !NO, TRY AGAIN.
        GO TO 1920
1950  . OUTPUT(IND) = 2048 !SILENT AREA, THEREFORE
                                  !SET OUTPUT = 2048 (OR 0).
```

```
      OUTPUT (IND + 1) = 0
      IF (IND.NE.255) GO TO 1970 !TIME TO DUMP OUTPUT BUF?
      WRITE(4'KVAR,END = 3000) (OUTPUT(KB),KB=1,256)
      IND = -1            !RESET POINTER
1970  IND = IND + 2
      GO TO 1920          !CONT. UNTIL END OF FILE.
C THIS AREA WILL BE DEALING WITH UNVOICED SEGMENTS WITH
C TSUM>NEWPI.
2000  GO TO 3000
2900  DO 2910 ILAX = K2,16
      DO 2905 JLAX = 1,256
      LATBUF(JLAX + ((ILAX-1)*256)) = 2048
2905  CONTINUE
2910  CONTINUE
      GO TO 1310
3000  ENDFILE 2
      ENDFILE 3
      ENDFILE 4
      STOP
      END
```

NONVO2

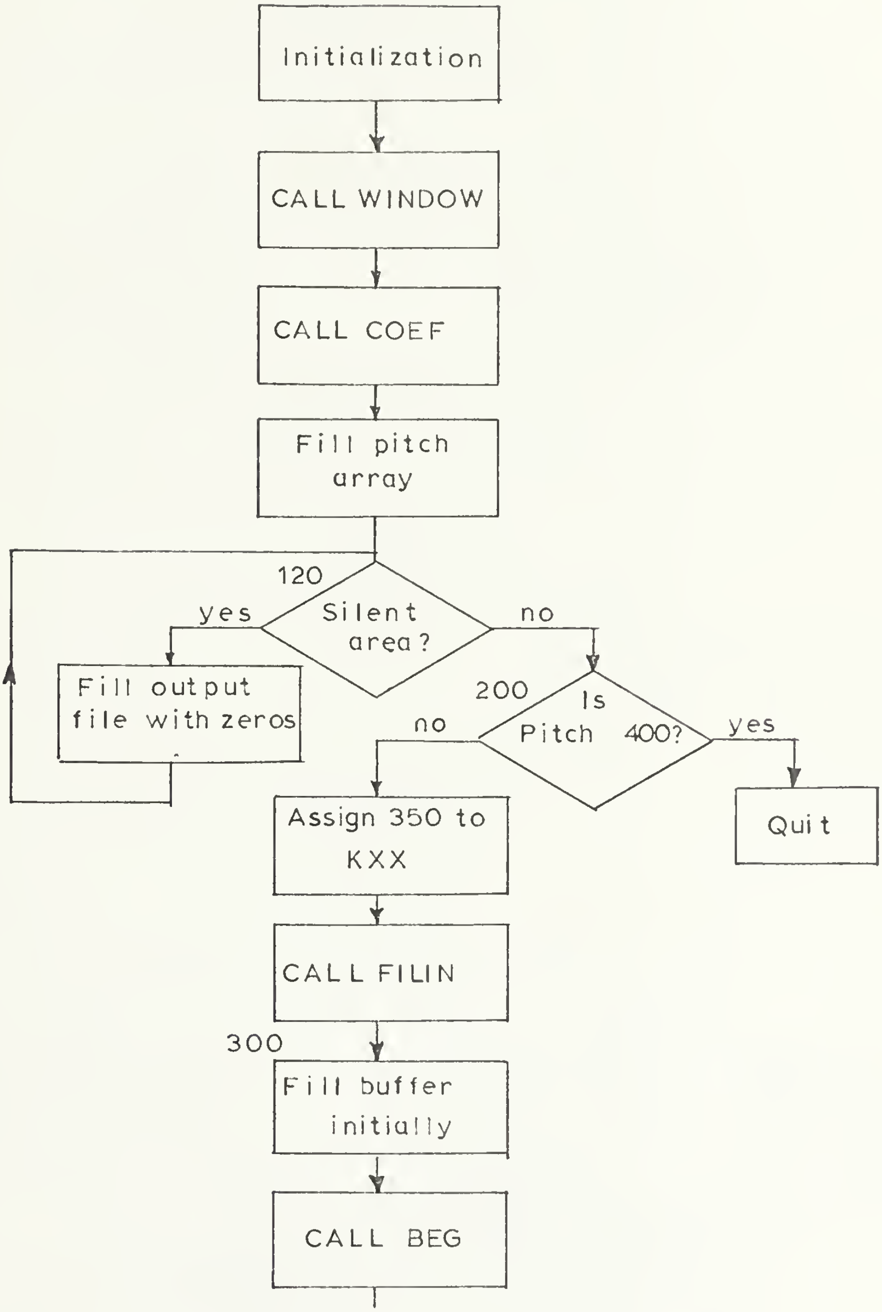


FIGURE A-15

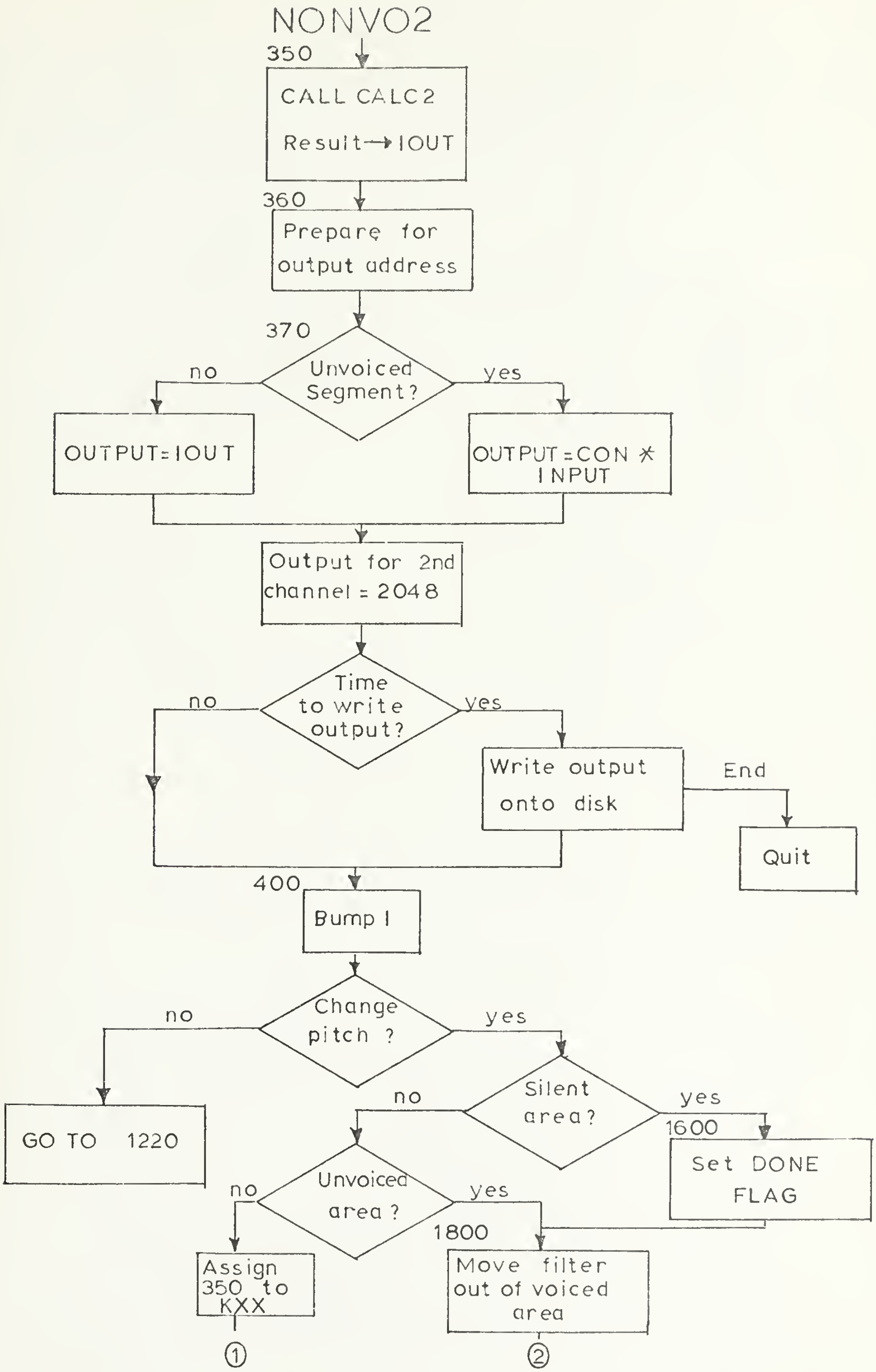


FIGURE A-15 (CON'T.)

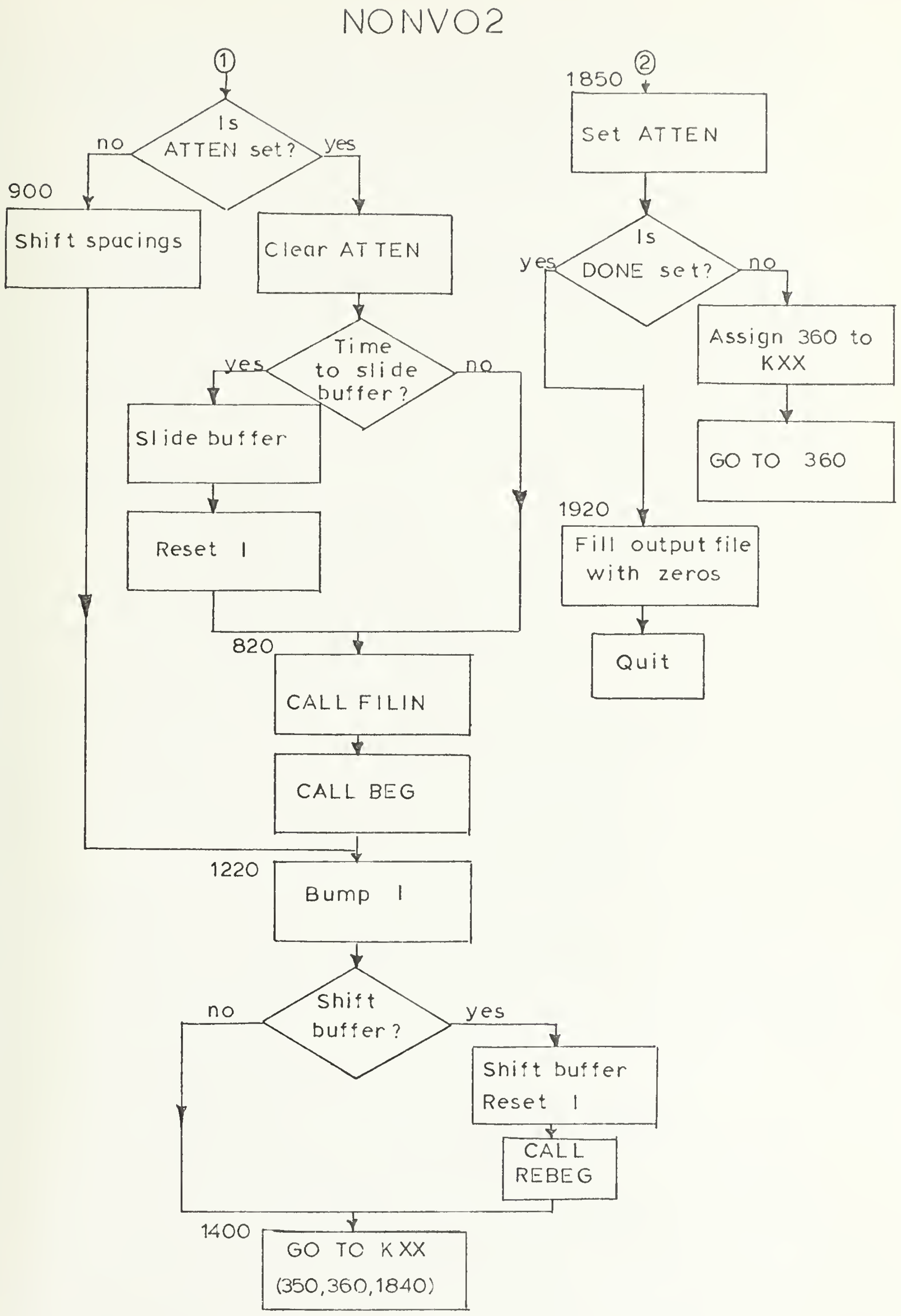


FIGURE A-15 (CON'T.)

```
C
C
C       TITLE:  NONV02.FOR      750324
C
C
C THIS PROGRAM IMPLEMENTS THE NONUNIFORMLY SPACED
C ADAPTIVE FILTER WITH THE CORRECTION FOR THE OVERLOAD
C PROBLEM.
C
C SUBROUTINES USED:
C       1.   WINDOW (NTYPE,K)
C       2.   COEF (NTYPE,K,A,I)
C       3.   FILIN (K,NEWPI,S,ISUM)
C       4.   REG (PSTART,S,I,K)
C       5.   CALC2 (A,DATBUF,S,I,K,IOUT,ISTART)
C       6.   REBEG (ISTART,I,K)
C
C ARRAYS:
C       1.   A - CONTAINS COEFFICIENTS.
C       2.   B - USED FOR RESCALED COFF. DURING OVERLOAD.
C       3.   DATBUF - CONTAINS INPUT.
C       4.   OUTPUT - TEMPORARY OUTPUT ARRAY.
C       5.   P - CONTAINS PITCH PERIODS.
C       6.   S - CONATINS SPACING FOR FILTER.
C
C VARIABLES:
```

```
C       1.  DONE -  LOGICAL DENOTES END OF SENTENCE.
C       2.  ATTEN - LOGICAL DENOTES UNVOICED AREAS.
C       3.  J - POINTER IN THE PITCH PERIOD ARRAY "P".
C       4.  ISUM - LENGTH OF THE FILTER.
C
C       INITIALIZATION
        DIMENSION A(15), B(15)
        INTEGER DATBUF(4096),OUTPUT(256),P(1000),S(14)
        INTEGER ISTART (15)
        LOGICAL DONE, ATTEN
        DATA DATBUF,OUTPUT,P,S/4095*0,256*0,1000*0,14*0/
        DATA ISTART, B/15*0,15*0.0/
        DATA IVAR,JVAR,KVAR/2,2,2/
        DATA DONE, ATTEN/.FALSE.,.FALSE./
        CALL ASSIGN (2,'RK1:SIGNAL.DAT')
        CALL ASSIGN (3,'RK1:PITCH.TAB')
        CALL ASSIGN (4,'RK1:OUTPUT.DAT')
        DEFINE FILE 2 (251,256,U,IVAR)
        DEFINE FILE 3 (1000,2,U,JVAR)
        CON = .3
C CHOOSE WINDOW TYPE
        CALL WINDOW (NTYPE,K)
C CALCULATE THE COEFFICIENTS
        CALL COEF (NTYPE,K,A,L)
C SPECIFY THE LENGTH OF THE OUTPUT FILE.
        LIMIT = 251 + (K*2) + 1
```

```
      DEFINE FILE 4 (LIMIT,256,U,KVAR)
C FILL THE PITCH ARRAY.
      DO 100 IA = 1,1000
      READ (3'JVAR,END = 110) P(IA)
100   CONTINUE
110   CONTINUE
      J = 1       !J IS THE POINTER IN THE PITCH PEROD ARRAY.
C THIS PART DETERMINES IF WE HAVE A SILENT AREA AT
C THE BEGINNING OF THE FILE.
120   IF (P(J).GE.0) GO TO 200          !SILENT AREA?
      M = IABS(P(J) * 2)        !YES, SET THE OUTPUT = 0.
      DO 150 KD = 1,(M-1),2
      KA = KD
130   IF (KA.LE.255) GO TO 140
      KA = KA - 256
      GO TO 130
140   OUTPUT (KA) = 2048
      OUTPUT(KA + 1) = 0
      IF (KA.NE.255) GO TO 150 !IS IT TIME TO WRITE A REC.?
      WRITE(4'KVAR) (OUTPUT(KB),KB=1,256) !YES.
150   CONTINUE
C BUMP THE COUNTER.
      J = J + 1
      GO TO 120
C PITCH PERIOD WAS NOT A SILENT ONE.
200   IF(P(J).GE.400) GO TO 3000
```

```
C VOICED, SET UP THE FILTER SPACING.
      ASSIGN 350 TO KXX
      CALL FILIN (K,P(J),S,ISUM)
C MEASUREMENTS SECTION
      IR = ISUM /256
      IF ((IR * 256).NE.ISUM) IR = IR + 1
C SET UP FOR INITIAL READING INTO BUFFER.
      IVAR = KVAR - IR
      DO 300 KE = 1,16
      READ(2'IVAR) (DATBUF(KF + ((KF-1)*256)),KF = 1,256)
300   CONTINUE
C SET UP THE BUFFER POINTER
      IF (KA.EQ.255) KA = -1   !RESET THE POINTER
                               !OF THE LAST OUTPUT.
      I = (IR * 256) + (KA + 2)
C INITIALIZE THE ISTART ARRAY
      CALL BEG (ISTART,S,I,K)
C DO THE OUTPUT CALCULATIONS
350   CALL CALC2 (A,DATBUF,S,I,K,IOUT,ISTART)
C COMPUTE THE ADDRESS FOR THE OUTPUT POINT.
360   IND = I
365   IF (IND.LE.255) GO TO 370 !DO WE HAVE A GOOD ADDRESS
      IND = IND - 256          !NO GO RECHECK.
      GO TO 365
370   IF(ATTEN) GO TO 375 !IS THIS A VOICED SEGMENT?
      OUTPUT(IND) = IOUT       !YES, GET CALCULATED VALUE.
```

```
      GO TO 380
C ATTENUATE THE INPUT.
375   OUTPUT(IND) = (CON * (DATBUF(I)-2048)) + 2048
380   OUTPUT(IND+1) = 0        !SECOND CHANNEL = 0.
C IS IT TIME TO DUMP THE OUTPUT BUFFER?
      IF (IND.NE.255) GO TO 400
      WRITE(4'KVAR,END=3000) (OUTPUT(KB),KB=1,256) !YES.
400   I = I + 1
C IS THERE A PITCH MARK?
      IF (DATBUF(I).LT.4075) GO TO 1220
500   NEWPI = P(J) * 2 !YES, GET A NEW VALUE OF PITCH.
      J = J + 1 !BUMP THE PITCH PERIOD POINTER.
C ARE WE ENTERING A SILENT AREA?
      IF (NEWPI.LT.0) GO TO 1600
C NO, ARE WE ENTERING AN UNVOICED AREA?
      IF (NEWPI.GT.400) GO TO 1800
C NO, THEREFORE WE ARE ENTERING A VOICED AREA.
C CHANGE THE FILTER.
      ASSIGN 350 TO KXX
C IS THE FILTER BEING INITIALIZED AFTER AN UNVOICED AREA?
      IF (ATTEN) GO TO 800
      GO TO 900 !THE FILTER IS NOT BEING RE-INITIALIZED.
800   ATTEN = .FALSE.  !CLEAR THE FLAG.
C TIME TO SLIDE THE BUFFER?
      IF (I.GT.(NEWPI*2*K)) GO TO 820
      JHOLD = I          !YES, SAVE THE POINTER.
```

```
        JHAP = NEWPI * 2 * K
        IRA = (JHAP - 1)/256
        IF ((IRA*256).NE.(JHAP-I)) IRA = IRA + 1
        IVAR = IVAR - 16 - IRA
        DO 810 IDY = 1,16
        READ (2'IVAR) (DATBUF(IDX + ((IDY-1)*256)),IDX=1,256)
810     CONTINUE
C RESET THE POINTER
        I = JHOLD + (256 * IRA)
820     CALL FILIN (K, (NEWPI/2), S,ISUM)
        CALL BEG (ISTART,S,I,K)
        GO TO 1220
900     KAA = 2 * K
1000    IF(KAA.EQ.1) GO TO 1100
        S(KAA) = S(KAA - 1)      !SHIFT THE SPACINGS
        KAA = KAA -1
        GO TO 1000
1100    S(1) = NEWPI !SET THE NEW PITCH PERIOD SPACING
        ISUM = 0 !COMPUTE THE LENGTH OF THE FILTER
        DO 1200 K1 = 1,2*K
        ISUM = ISUM + S(K1)
1200    CONTINUE
        CALL BEG (ISTART,S,I+1,K)
1220    I = I + 1
C IS IT TIME TO SHIFT THE BUFFER?
        IF (I.LT.4096) GO TO 1400
```

```
C YES, SHIFT THE BUFFER.
      IR = ISUM / 256
      IF ((IR*256).NE.ISUM) IR = IR + 1
      IVAR = IVAR - IR
      DO 1300 K2 = 1,16
      IF (IVAR.GE.252) GO TO 2900
      READ (2'IVAR) (DATBUF(K3 + ((K2-1)*256)),K3=1,256)
1300  CONTINUE
C RESET I
1310  I = (IR * 256) + 1
C RESET THE ISTART ARRAY.
      CALL REBEG (ISTART,1,K)
1400  GO TO KXX, (350,360,1840)
C THIS AREA RESERVED FOR SILENT AREA LATER.
1600  NEWPI = IABS(NEWPI)      !GET RID OF MINUS SIGN
      DONE = .TRUE.            !SET DONE FLAG.
      GO TO 1800
C UNVOICED AREA
      I = I + 1          !BUMP THE POINTER.
      IF (NEWPI.LE.ISUM) GO TO 2000
1820  K7 = 1             !SET UP A COUNTER.
1825  KAB = 0
C ARE WE THROUGH SHIFTING?
1830  IF (KAB.EQ.((2*K)-K7)) GO TO 1835
      KNEW = 2 * K - KAB
      KOLD = KNEW - 1
```

```
        S(KNEW) = S(KOLD)          !NO,CONTINUE
        KAB = KAB + 1
        GO TO 1830
1835    ISUM = 0
        DO 1836 ITE = 1,2*K+1
        ISUM = ISUM + S(ITE)
1836    CONTINUE
        CALL BEG (ISTART,S,I+1,K)
        K8 = 1
1837    ASSIGN 1840 TO KXX
        GO TO 350 !GO COMPUTE AN OUTPUT.
1840    K8 = K8 + 2
        IF (K8.LT.S(1)) GO TO 1837      !ARE WE THRU?
        K7 = K7 + 1          !YES, BUMP K7
        IF (K7 - (2*K)) 1825,1835,1850 !ARE WE THRU?
C ENTER THE ATTENUATION PROCEDURE
1850    ATTEN = .TRUE.     !SET THE ATTENUATION FLAG.
        IF (DONE) GO TO 1900       !ARE WE ALMOST THROUGH?
        ASSIGN 360 TO KXX !NO, THIS IS ONLY AN UNVOICED AREA.
        GO TO 360
1900    IND = 1
C DO WE HAVE A SUITABLE IND?
1920    IF (IND.LE.255) GO TO 1950
        IND = IND - 256            !NO, TRY AGAIN.
        GO TO 1920
1950    OUTPUT(IND) = 2048         !SILENT AREA, THEREFORE
```

```
                                  !SET OUTPUT = 2048 (OR 0).
      OUTPUT (IND + 1) = 0
      IF (IND.NE.255) GO TO 1970 !TIME TO DUMP OUTPUT BUF?
      WRITE(4'KVAR,END = 3000) (OUTPUT(KB),KB=1,256)
      IND = -1            !RESET POINTER
1970  IND = IND + 2
      GO TO 1920          !CONT. UNTIL END OF FILE.
C THIS AREA WILL BE DEALING WITH UNVOICED SEGMENTS WITH
C ISUM>NEWPT.
2000  WRITE(7,2200)
2200  FORMAT (' ','PROGRAM EXITED FROM 2000')
      GO TO 3000
2900  DO 2910 ILAX = K2,16
      DO 2905 JLAX = 1,256
      DATBUF(JLAX + ((ILAX-1)*256)) = 2048
2905  CONTINUE
2910  CONTINUE
      GO TO 1310
3000  ENDFILE 2
      ENDFILE 3
      ENDFILE 4
      STOP
      END
```

BEG

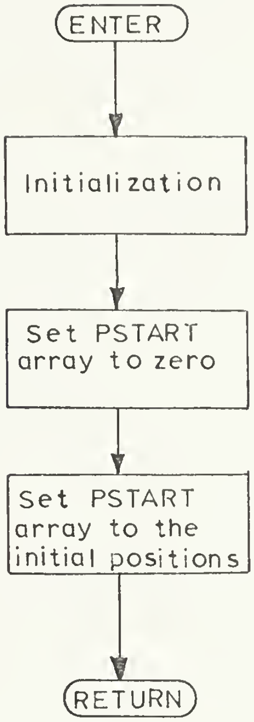


FIGURE A-16

```
C
C       TITLE:  BEG.FOR  750323
C
C
C THIS PROGRAM IS A SUBROUTINE FOR THE NONVO2.FOR PROGRAM.
C IT INITIALIZES THE ISTART ARRAY WHEN CALLED.
C
C       INITIALIZATION
        SUBROUTINE BEG (PSTART,S,I,K)
        INTEGER PSTART(15), S(14)
C ZERO THE PSTART ARRAY
        DO 10 IU = 1,2*K + 1
        PSTART(IU) = 0
10      CONTINUE
C INITIALIZE THE ARRAY WITH THE SPACINGS.
        KND = I
        PSTART (1) = I
        DO 20 J = 2,(2*K) + 1
        KND = KND - S(J - 1)
        PSTART (J) = KND
20      CONTINUE
        RETURN
        END
```

REBEG

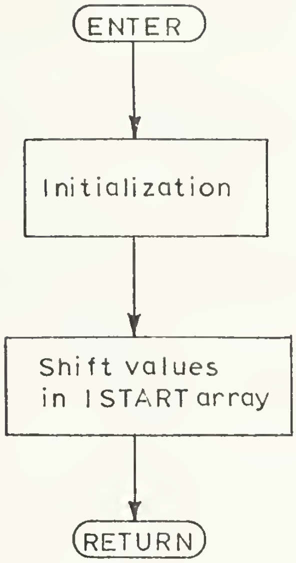


FIGURE A-17

```
C
C
C       TITLE:  REPEG.FOR       750326
C
C THIS SUBROUTINE RE-INITIALIZES THE ISTART ARRAY AFTER
C A SHIFT IN THE BUFFER HAS BEEN PERFORMED.
C
C       INITIALIZATION
        SUBROUTINE REPEG (ISTART,I,K)
        INTEGER ISTART(15)
C PERFORM THE EXCHANGE OF THE ELEMENTS IN ISTART.
        DO 20 J = 1, 2*K + 1
        ISTART  (J) = I - (4096 - ISTART(J))
20      CONTINUE
        RETURN
        END
```

CALC2

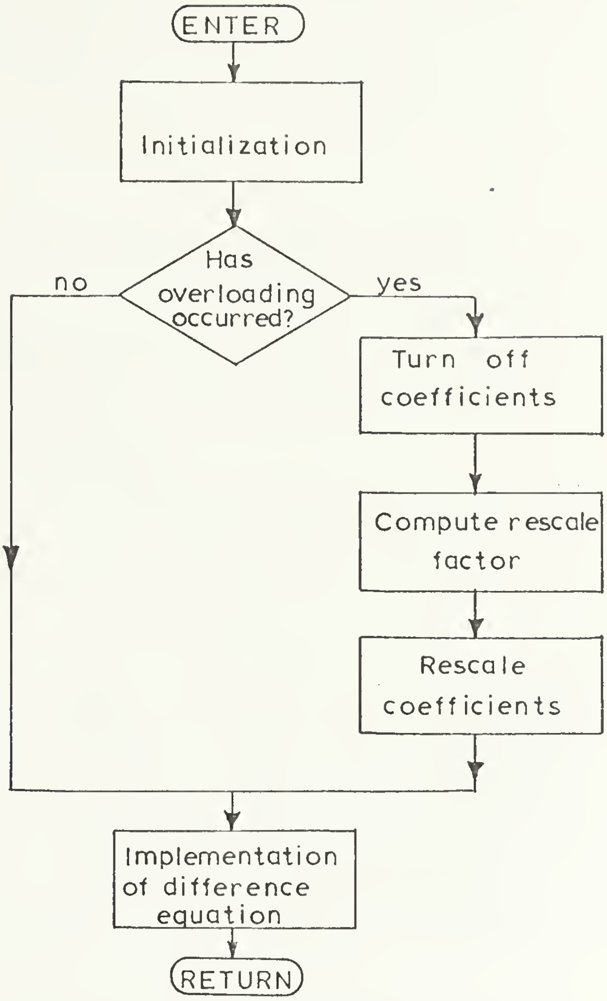


FIGURE A-18

```
C
C       TITLE:  CALC2.FOR        750324
C
C THIS SUBROUTINE IS USED WITH THE MAIN PROGRAM NONVO2
C AND IS USED TO CALCULATE AN OUTPUT POINT.  IT ALSO
C DETERMINES IF THE COEFFICIENTS NEED TO BE "TURNED OFF".
C PARAMETERS:
C ARRAYS:
C       1.  A - CONTAINS THE COEFFICIENTS
C       2.  DATBUF - CONTAINS THE INPUT DATA
C       3.  S - CONTAINS THE SPACINGS BETWEEN THE
C             COEFFICIENTS.
C       4.  ISTART - CONTAINS THE INITIAL LOCATIONS
C             OF THE COEF. FOR THE PRESENT FILTER.
C VARIABLES:
C       1.  J - POINTER OF THE CURRENT PROCESSING IN DATBUF
C       2.  K - VALUE THAT DETERMINES THE NO. OF COEF.
C       3.  IOUT - A VALUE RETURNED TO THE MAIN PROGRAM.
C             THE OUTPUT OF THE FILTER FROM ONE POINT.
C
        SUBROUTINE CALC2 (A,DATBUF,S,J,K,IOUT,ISTART)
        DIMENSION A(15), B(15)
        INTEGER DATBUF (4096),S(14),ISTART(15)
        LOGICAL SCALE
        SCALE = .FALSE.
C SEE IF ANY BAD SAMPLES HAVE BEEN REACHED.
```

```
      JND = 1
      B(1) = A(1)
      DO 10 IP = 2,(2*K) + 1
      JND = JND - S(IP-1)
C DO WE HAVE A COEFFICIENT THAT HAS PASSED ITS
C AREAS?
      IF (JND.GE.ISTART(IP-1)) GO TO 8
      B(IP) = A(IP)
      GO TO 10
8     B(IP) = 0.0          !YES, TURN OFF THE COEF.
      SCALE = .TRUE.
10    CONTINUE
      IF (.NOT.SCALE) GO TO 18 !DO WE NEED TO RESCALE?
      SCALE = .FALSE.  !YES, RESCALE.
      SCAL = 0.0
C COMPUTE THE RESCALE FACTOR
      DO 15 IT = 1,(2*K) + 1
      SCAL = SCAL + B(IT)
15    CONTINUE
C RESCALE THE COEFFIECIENTS
      DO 17 IU = 1,(2*K) + 1
      B(IU) = B(IU) / SCAL
17    CONTINUE
C COMPUTAITON OF THE OUTPUT POINT
18    Y = 0.0
      Y = B(1) * DATBUF(1)
```

```
      INDEX = I
      DO 20 IY = 2,(2*K) + 1
      INDEX = INDEX - S(IY -1)
      Y = Y + (B(IY)* DATBUF(INDEX))
20    CONTINUE
      IOUT = IFIX (Y + .5)
      RETURN
      END
```

Thesis
F7868

163417

Frazier
An adaptive filtering approach toward speech enhancement.